PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F — A2

(11) International Publication Number: **WO 99/44113**

(43) International Publication Date: 2 September 1999 (02.09.99)

(21) International Application Number: PCT/US99/04247

(22) International Filing Date: 25 February 1999 (25.02.99)

(30) Priority Data:
09/030,697 25 February 1998 (25.02.98) US

(71) Applicant *(for all designated States except US)*: LEXAR MEDIA, INC. [US/US]; 47421 Bayside Parkway, Fremont, CA 94538 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: ESTAKHRI, Petro [US/US]; 7966 Foothills Knolls, Pleasanton, CA 94566 (US). IMAN, Berhanu [US/US]; 946 Iris Avenue, Sunnyvale, CA 94086 (US).

(74) Agents: IMAM, Maryam et al.; Oppenheimer Wolff & Donnelly LLP, Suite 400, 101 Park Center Plaza, San Jose, CA 95113 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: INCREASING MEMORY PERFORMANCE IN FLASH MEMORY DEVICES BY PERFORMING SIMULTANEOUS WRITE OPERATION TO MULTIPLE DEVICES

(57) Abstract

The present invention includes a digital system having a controller semiconductor device coupled to a host and a nonvolatile memory bank including a plurality of nonvolatile memory devices. The controller transfers information, organized in sectors, with each sector including a user data portion and an overhead portion, between the host and the nonvolatile memory bank and stores and reads two bytes of information relating to the same sector simultaneously within two nonvolatile memory devices. Each nonvolatile memory device is defined by a row of memory locations wherein corresponding rows of at least two semiconductor devices maintain two sectors of information therein with the overhead information relating to the two sectors maintained in one of the memory rows of the nonvolatile memory device. Each 32 sectors of information defines a block identified by a virtual physical block address with a block of information expanding between two memory devices wherein an even and an odd byte of a sector is simultaneously read from or written to two nonvolatile memory devices. In another embodiment, the controller stores an entire sector of information within a single nonvolatile memory device and reads from or writes to, a sector of information by processing corresponding bytes of at least two sectors in two nonvolatile memory devices simultaneously.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

Specification

INCREASING MEMORY PERFORMANCE IN FLASH MEMORY DEVICES BY PERFORMING SIMULTANEOUS WRITE OPERATION TO MULTIPLE DEVICES

BACKGROUND OF THE INVENTION

Cross Reference to Related Applications

This application is a continuation-in-part of our prior application Serial No. 08/946,331 filed October 7, 1997, entitled " Moving Sequential Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", which is a continuation-in-part of application Serial No. 08/831,266, filed March 31, 1997, entitled "Moving Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture".

Field of the Invention

This invention relates to the field of digital systems, such as personal computers and digital cameras, employing nonvolatile memory as mass storage, for use in replacing hard disk storage or conventional film. More particularly, this invention relates to an architecture for increasing the performance of such digital systems by increasing the rate at which digital information is read from and written to the nonvolatile memory.

Description of the Prior Art

With the advent of higher capacity solid state storage devices (nonvolatile memory), such as flash or EEPROM memory, many digital systems have replaced conventional mass storage devices with flash and/or EEPROM memory devices. For example, personal computers (PC s) use solid state storage devices for mass storage purposes in place of

conventional hard disks. Digital cameras employ solid state storage devices in cards to replace conventional films.

Fig. 1 shows a prior art memory system 10 including a controller 12, which is generally a semiconductor (or integrated circuit) device, coupled to a host 14 which may be a PC or a digital camera. The controller 12 is further coupled to a nonvolatile memory bank 16. Host 14 writes and reads information, organized in sectors, to and from memory bank 16 which includes a first nonvolatile memory chip 18 and a second nonvolatile memory chip 20. Chip 18 includes: an I/O register 22 having a port 24 connected to a port 26 of controller 12 via a first bus 28 which includes 8 bit lines; and a storage area 30 coupled with I/O register 22. Chip 20 includes: an I/O register 32 having a port 34 connected to a port 36 of controller 12 via a second bus 38 which includes 8 bit lines; and a storage area 40 coupled with I/O register 32. The first and second buses 28, 38 are used to transmit data, address, and command signals between the controller and the memory chips 18 and 20. The least significant 8 bits (LSBs) of 16 bits of information are provided to chip 18 via the first bus 28, and the most significant 8 bits (MSBs) are provided to the chip 20 via the second bus 38.

Memory bank 16 includes a plurality of block locations 42 each of which includes a plurality of memory row locations. Each block location of the memory bank is comprised of a first sub-block 44 located in the first non-volatile memory chip, and a corresponding second sub-block 46 located in the second non-volatile memory chip. Each memory row location includes a first row-portion 48 and a corresponding second row-portion 50. In the depicted embodiment each of the first and second row-portions 48 and 50 includes storage for 256 bytes of data information plus an additional 8 bytes of storage space for overhead information. Where a sector includes 512 bytes of user data and 16 bytes of non-user data (the latter commonly referred to as overhead information), 256 bytes of the user data and 8 bytes of the overhead information of the sector may be maintained in the first row portion 48 of chip 18 and the remaining 256 bytes of user data and remaining 8 bytes of overhead information of the same sector may be maintained in the second row portion 50 of chip 20. Thus, half of a sector is stored in a memory row location 48 of chip 18 and the other half of the sector is stored in memory row location 50 of chip 20. Additionally, half of the overhead information of each stored sector is maintained by chip 18 and the other half by chip 20.

In general, reading and writing data to flash memory chips 18 and 20 is time consuming. Writing data to the flash memory chips is particularly time consuming because

data must be latched in I/O registers 22 and 32, which are loaded 1 byte at a time via the first and second buses, and then transferred from the I/O registers 22 and 32 to the memory cells of the flash memory chips 18 and 20 respectively. The time required to transfer data from the I/O registers to memory, per byte of data, is proportional to the size of the I/O registers and the size of the flash memory chip.

During a write operation, controller 12 writes a single sector of information to memory bank 16 by: (1) transmitting a write command signal to each of chips 18 and 20 via buses 28 and 38 simultaneously; (2) transmitting address data to chips 18 and 20 specifying corresponding sub-blocks 44 and 46 of the chips via buses 28 and 38 simultaneously; and (3) sequentially transmitting a byte of user data to each of chips 18 and 20 via buses 28 and 38 simultaneously for storage in the corresponding sub-blocks 44 and 46. The problem with such prior art systems is that while two bytes of information are written and read at a time, only one sector of information is accommodated at a time by the memory bank 16 during a write command initiated by the host 14.

Another prior art digital system 60 is shown in Fig. 2 to include a controller 62 coupled to a host 64, and a nonvolatile memory bank 66 for storing and reading information organized in sectors to and from nonvolatile memory chip 68, included in the memory bank 66. While not shown, more chips may be included in the memory bank, although the controller, upon command by the host, stores an entire sector in one chip. A block, such as block 0, includes 16 sectors S0, S1,..., S15. Also included in the chip 68 is an I/O register 70, which includes 512 bytes plus 16 bytes, a total of 528 bytes, of storage space. The controller transfers information between host 64 and memory 66 a byte at-a-time. A sector of 512 bytes of user data plus 16 bytes of overhead information is temporarily stored in the I/O register during a write operation and then transferred to one of the blocks within the memory device for storage thereof. During a read operation, a sector of information is read from one of the blocks of the memory device and then stored in the I/O register for transfer to the controller . An important problem with the prior art architecture of Fig. 2 is that while a total of 528 bytes may be stored in the I/O register 36, only one byte of sector information may be transferred at a time between the controller and the memory bank thereby impeding the overall performance of the system.

Both of the prior art systems of Figs. 1 and 2 maintain LBA to PBA mapping information for translating a host-provided logical block address (LBA) identifying a sector

of information to a physical block address (PBA) identifying the location of a sector within the memory bank. This mapping information may generally be included in volatile memory, such as a RAM, within the controller, although it may be maintained outside of the controller.

Fig. 3 shows a table diagram illustrating an example of an LBA-PBA map 300 defined by rows and columns, with each row 302 being uniquely identified, addressed, by a value equal to that of the LBA received from the host divided by 16. The row numbers of Fig. 3 are shown using hexadecimal notation. Thus, for example, row 10H (in Hex.) has an address value equal to 16 in decimal. Each row 302 of map 300, includes a storage location field 304 for maintaining a virtual PBA value, an 'old' flag field 306, a 'used' flag field 308, and a 'defect' flag field 310. The flag fields provide information relating to the status of a block of information maintained within the memory bank (in Figs. 1 and 2). The virtual PBA field 304 stores information regarding the location of the block within the memory bank.

Fig. 4 shows a table diagram illustrating an exemplary format for storage of a sector of data maintained in a memory bank. The virtual PBA field 304 (Fig. 3) provides information regarding the location of a block 400 of information with each block having a plurality of sectors 402. Each sector 402 is comprised of a user data field 404, an ECC field 406, an 'old' flag field 408, a 'used' flag field 410 and a 'defect' flag field 412.

A further problem associated with prior art systems of the kind discussed herein is that the table 300 (in Fig. 3) occupies much 'real estate' and since it is commonly comprised of RAM technology, which is in itself costly and generally kept within the controller, there is substantial costs associated with its manufacturing. Furthermore, as each row of table 300 is associated with one block of information, the larger the number of blocks of information, the larger the size of the table, which is yet an additional cost for manufacturing the controller and therefore the digital system employing such a table.

What is needed is a digital system employing nonvolatile memory for storage of digital information organized in sector format for reducing the time associated with performing reading and writing operations on the sectors of information thereby increasing the overall performance of the system while reducing the costs of manufacturing the digital system.

SUMMARY OF THE INVENTION

It is an object of the present invention to increase the performance of a digital system having a controller coupled to a host for operating a nonvolatile memory bank including one or more nonvolatile memory devices, such as flash and/or EEPROM chips, by reducing the time associated with reading and writing information to the nonvolatile memory bank.

It is another object of the present invention, as described herein, to decrease the time associated with storing sectors of information by writing at least two sectors of information to at least two nonvolatile memory semiconductor devices during a single write command initiated by the host.

It is another object of the present invention as described herein to decrease the time associated with reading sectors of information by reading at least two sectors of information from at least two nonvolatile memory semiconductor devices during a single read command initiated by the host.

It is a further object of the present invention to store overhead information associated with two sectors of information in one of the two nonvolatile memory semiconductor devices.

It is yet another object of the present invention to simultaneously access two bytes of a sector of information stored within two nonvolatile memory devices thereby increasing the rate of performance of a system employing the present invention by an order of magnitude of at least two.

It is yet another object of the present invention to access one byte of a first sector and one byte of a second sector of information simultaneously within two nonvolatile memory devices thereby increasing the rate of performance of a system employing the present invention.

It is a further object of the present invention to reduce the size of a volatile memory table, or map, that maintains translations between the host-provided sector addresses to addresses of blocks within the nonvolatile memory devices thereby reducing the cost of manufacturing the digital system.

Briefly, the present invention includes a digital system having a controller semiconductor device coupled to a host and a nonvolatile memory bank including a plurality of nonvolatile memory devices. The controller transfers information, organized in sectors,

with each sector including a user data portion and an overhead portion, between the host and the nonvolatile memory bank and stores and reads two bytes of information relating to the same sector simultaneously within two nonvolatile memory devices. Each nonvolatile memory device is defined by a row of memory locations wherein corresponding rows of at least two semiconductor devices maintain two sectors of information therein with the overhead information relating to the two sectors maintained in one of the memory rows of the nonvolatile memory device. Each 32 sectors of information defines a block identified by a virtual physical block address with a block of information expanding between two memory devices wherein an even and an odd byte of a sector is simultaneously read from or written to two nonvolatile memory devices. In another embodiment, the controller stores an entire sector of information within a single nonvolatile memory device and reads from or writes to, a sector of information by processing corresponding bytes of at least two sectors in two nonvolatile memory devices simultaneously.

These and other objects and advantages of the present invention will no doubt become apparent to those skilled in the art after having read the following detailed description of the preferred embodiments illustrated in the several figures of the drawing.

IN THE DRAWINGS

Fig. 1 is a block diagram of a prior art memory system in which a single sector of information is written, two bytes at a time during a write operation, to a memory bank including two memory units each having capacity to store 256 bytes of user data in a single row location;

Fig. 2 is a block diagram of a prior art memory system in which a single sector of information is written, one byte at a time during a write operation, to a memory bank including at least one memory unit having capacity to store 512 bytes of user data in a single row location;

Fig. 3 is a table diagram illustrating an exemplary map for translating a host-provided logical block address (LBA) identifying a sector of information to a physical block address (PBA) identifying a location for the sector within a memory bank;

Fig. 4 is a table diagram illustrating an exemplary format for storage of a sector of data maintained in a memory bank;

Fig. 5 is a generalized block diagram of a memory system in accordance with the present invention in which two sectors of information are written, two bytes at a time during a single write operation, to a memory bank including at least two memory units each having capacity to store 512 bytes of user data in a single row location;

Figure 6 is a detailed block diagram of the memory system of Fig. 5;

Fig. 7 is a table diagram generally illustrating a memory storage format for storing a block, including 32 sectors, of information in a memory bank including two non-volatile memory units wherein an even sector and an odd sector are stored in a single memory row location and wherein even data bytes of both sectors are stored in a row portion located in a first of the memory units and odd data bytes of both sectors are stored in a second row portion located in the second of the memory units;

Fig. 8A is a table diagram generally illustrating organization of an exemplary LBA-PBA map for use in accordance with the present invention;

Fig. 8B shows a block diagram illustrating formats of address information identifying sectors and associated blocks of information in accordance with the present invention;

Fig. 9 is a timing diagram illustrating the timing of control, address, and data signals for a write operation performed by the memory system of Fig. 6 wherein two sectors of information are simultaneously written, during a single write operation, to a memory bank having the memory storage format illustrated in Fig. 7;

Fig. 10 is a table diagram illustrating a memory bank having a memory storage format as depicted in Fig. 7 wherein a single sector is written to a particular memory row location of the memory bank;

Fig. 11 is a table diagram illustrating a memory bank having an alternative memory storage format as depicted in Fig. 7 wherein a single sector is written to a particular memory row location of the memory bank;

Fig. 12 is a flowchart illustrating a process of simultaneously writing two sectors of information to two memory units during a single write operation in accordance with the present invention;

Fig. 12a shows a flow chart of the steps performed in executing the defect management routine of Fig. 12.

Fig. 13 is a table diagram generally illustrating an alternative memory storage format for storing a block, including 32 sectors, of information in a memory bank including two non-volatile memory units wherein an even sector and an odd sector are stored in a single memory row location and wherein an even sector is stored in a first row portion located in a first of the two memory units and an odd sector is stored in a second row portion located in the second of the two memory units;

Fig. 14 shows a timing diagram illustrating the timing of control, address, and data signals for a process of erasing a block of a memory bank in accordance with principles of the present invention; and

Fig. 15 is a flowchart illustrating a process of erasing a block, including a first sub-block stored in a first memory unit and a second sub-block stored in a second memory unit, in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 5 shows a generalized block diagram at 500 of a memory system in accordance with principles of the present invention. The system includes a memory card 502 coupled to a host system 504. In one embodiment, host 504 is a digital camera and memory card 502 is a digital film card, and in another embodiment, host 504 is a personal computer system and memory card 502 is a PCMCIA card. Memory card 502 includes: a non-volatile memory bank 506 including a plurality of non-volatile memory units 508 for storing sectors of information organized in blocks; a memory controller 510 coupled to the memory bank via a memory bus 512, and coupled to the host 504 via a host bus 514. Memory controller 510 controls transfer of sector-organized information between host 504 and memory bank 506. Each sector of information includes a user data portion and an overhead portion. The memory controller performs write and read operations, in accordance with the present invention, to and from the memory units of the memory bank as further explained below.

In the present invention, the non-volatile memory bank 506 may include any number of non-volatile memory units 508 while in a preferred embodiment, the non-volatile memory bank has an even number of memory units. Also in the preferred embodiment, each of the non-volatile memory units is a flash memory integrated circuit device.

Fig. 6 shows a detailed block diagram at 600 of the memory system of Fig. 5. Controller 510 is shown to include: a host interface 610 connected to the host 504 via host bus 514 for transmitting address, data, and control signals between the controller and the host; a data buffer 614 having a port 616 coupled to a port 618 of the host interface; a microprocessor 620 having a port 622 coupled to a port 624 of the host interface; a code storage unit 626 having a port 628 coupled to a port 630 of the microprocessor; a boot ROM unit 632 having a port 634 coupled to port 630 of the microprocessor and to port 628 of the code storage unit; a space manager 636 having a port 638 coupled to a port 640 of the microprocessor; a flash state machine 642 including a port 644 coupled to a port 646 of the microprocessor, a port 648 coupled to a port 650 of the space manager, and a port 645 coupled to a port 647 of the data buffer; a memory input/output unit 652 having a port 654 coupled to a port 656 of the flash state machine; an error correction code logic unit (ECC logic unit) 660 having a port 662 coupled to a port 664 of the flash state machine, and a port 666 coupled to a port 668 of the data buffer 614.

In the depicted embodiment, memory bank 506 includes two non-volatile memory units (although additional memory units may be included, only two are shown for simplicity); a first flash memory chip 670 designated FLASH0 and a second flash memory chip 672 designated FLASH1. First flash memory chip 670 includes a first input/output register (first I/O register) 671 and a storage area 669. Second flash memory chip 672 includes a second input/output register (second I/O register) 673 and a storage area 674.

Memory bus 512 is used to transmit address, data, and control signals between the controller 510 and memory bank 506. Memory bus 512 includes a flash bus 675 connected to a port 676 of memory I/O unit 652 for transmitting address, data, and command signals between flash memory chips 670, 672 and the memory I/O unit 652. Flash bus 675 includes 16 bit lines, 8 bit lines of which form a first bus 680 connected to a port 682 of I/O register 671 of the first flash memory chip, and another 8 bit lines of which form a second bus 684 connected to a port 686 of I/O register 673 of the second flash memory chip.

Memory bus 512 also includes: a control bus 690 which connects a control signal (CTRL signal) output 692 of the flash state machine 642 to an input 694 of the first flash memory chip and to an input 696 of the second flash memory chip; a chip enable line 698 which connects a chip enable (CE) output 700 of the flash state machine 642 to an enable input 702 of the first flash memory chip and to enable an input 704 of the second flash

memory chip; and a ready/busy signal (FRDY-BSY* signal) line 706 which connects an output 708 of the first flash memory chip and an output 710 of the second flash memory chip to an input 712 of the flash state machine 642.

Microprocessor 620, at times (for example, during initialization of the memory system), executes program instructions (or code) stored in ROM 632, and at other times, such as during operation of the memory system, the microprocessor executes code that is stored in code storage unit 626, which may be either a volatile, i.e., read-and-write memory (RAM) or a non-volatile, i.e., EEPROM, type of memory storage. Prior to the execution of program code from code storage unit 626, the program code may be stored in the memory bank 506 and later downloaded to the code storage unit for execution thereof. During initialization, the microprocessor 620 can execute instructions from ROM 632.

Sector-organized information, including user data and overhead information, is received at host interface 610 from host 504 via host bus 514 and provided to the data buffer 614 for temporary storage therein. Sectors of information stored in the data buffer are retrieved under control of flash state machine 642 and provided to memory bank 506 in a manner further described below. It is common in the industry for each sector to include 512 bytes of user data plus overhead information. Although a sector may include other numbers of bytes of information, in the preferred embodiment, a sector has 512 bytes of user data and 16 bytes of overhead information.

ECC logic block 660 includes circuitry for performing error coding and correction on the sector-organized information. ECC logic block 660 performs error detection and/or correction operations on the user data portions of each sector stored in the flash memory chips 670, 672 or data received from host 504.

When required, the space manager 636 finds a next unused (or free) non-volatile memory location within the memory bank for storing a block of information with each block including multiple sectors of information. In the preferred embodiment, a block includes 32 sectors although, alternatively a block may be defined to include another number of sectors such as, for example, 16. The physical address of a storage block located within memory bank 506, referred to as a virtual physical block address (virtual PBA), and the physical block address of a sector of information located within the memory bank 506, referred to as an actual physical block address (actual PBA), is determined by the space manager by performing a translation of a logical block address (LBA) received from the host. An actual

LBA received from host 504 (a host-provided LBA) identifies a sector of information. Space manager 636 includes a space manager memory unit, which is preferably a volatile memory unit, for storing an LBA-PBA map for translating a modified version of the host-provided LBAs to virtual PBAs as further explained below. In the depicted embodiment, the space manager includes a space manager RAM unit (SPM RAM unit) 720 for storing the LBA-PBA map under the control of a space manager controller (SPM controller) 724 which is coupled to the SPM RAM unit.

Fig. 7 shows a table diagram generally illustrating organization of user data, error correction information, and flag information stored in memory bank 506 in accordance with an embodiment of the present invention. Memory bank 506 includes a plurality of M blocks 727 designated BLCK0, BLCK1, BLCK(M-1), each having a virtual physical block addresses (PBA). Each of the blocks 727 includes a plurality of N memory row locations 728 designated ROW0, ROW1,...ROW15 where, in the preferred embodiment, N=16. Each block 727 of memory bank 506 is comprised of a first sub-block 730 of first flash memory chip 670, and a corresponding second sub-block 731 of second flash memory chip 672. Corresponding sub-blocks 730, 731, which together form a block, are identified by the same virtual PBA. Each memory row location 728 includes a first row-portion 732 and a corresponding second row-portion 733 In the depicted embodiment each of the first and second row-portions 732, 733 includes storage for 512 bytes of data information plus additional storage space for other information. In the depicted embodiment, the storage of information in the first row-portions 732 of the first flash memory chip is accomplished in a manner dissimilar from that in the second row-portions 733 of the second flash memory chip.

Each of the first row-portions 732 includes: a first even sector field 734 for storing even data bytes D0, D2, D4,...D510 of an even sector (S0, S2, S4,...) of information; a first spare field 736; a first odd sector field 738 for storing even data bytes D0, D2, D4,...D510 of an odd sector (S1, S3, S5,...) of data; and a second spare field 740. Each of the second row-portions 733 includes: a second even sector field 742 for storing odd data bytes D1, D3, D5,...D511 of the even sector of data which has it's corresponding even data bytes stored in first even sector field 734; a first error correction field 744 for storing error correction information corresponding to the even sector of information stored collectively in fields 734 and 742; a second odd sector field 746 for storing odd data bytes of the odd sector of information which has it's even data bytes stored in first odd sector field 738; a second error

correction field 748 for storing ECC information corresponding to the odd sector of information stored collectively in fields 738 and 746; a block address field 750; and a flag field 752. Fields 734 and 742 form an even sector location while fields 738 and 746 form an odd sector location. It is understood in the present invention that fields 734 and 742 could alternatively form an odd sector location while fields 738 and 746 could alternatively form an even sector location, and that fields 734 and 738 could alternatively be used to store odd data bytes while fields 742 and 746 could alternatively be used to store even data bytes. Additionally, first row-portion 732 could alternatively be used for storing the overhead information relating to the sectors stored in the memory row location 728.

Flag field 752 is used for storing flag information which is used by controller 510 (Fig. 6) during access operations as further explained below. Block address field 750 is used for storing a modified version of a host-provided LBA value which is assigned to a block, as further described below. Only a single block address entry is required in the block address field per block. In a preferred embodiment, a modified host-provided LBA value is entered in block address field 759 of the Nth row, ROW15, of the row locations 728 of each block 727.

In operation, the controller 510 (Fig. 6) accesses an even sector of information stored collectively in the first and second flash memory chips by simultaneously accessing first and second even sector fields 734, 742 of corresponding row-portions of the first and second flash memory chips via the first and second split buses 680, 684 (Fig. 6), respectively. The first and second split buses 680, 684 (Fig. 6) include lines coupled to receive the even and odd data bytes respectively of a sector of information. The controller 510 (Fig. 6) accesses an odd sector of information stored collectively in the first and second flash memory chips by simultaneously accessing the first and second odd sector fields 738, 746 via the first and second split buses 680, 684 (Fig. 6), respectively. The split buses 680, 684 (Fig. 6) also provide for: transmission of ECC information between the flash memory chips and the flash state machine 642 and ECC logic unit 660 of the memory controller 510; and transmission of address information from flash state machine 642 to the flash memory chips.

Controller 510 (Fig. 6) monitors the status of blocks 727 of memory bank 506 using the space manager 636. In one embodiment, controller 510 (Fig. 6) monitors the status of each block location 727 of the memory bank using block level flags including a used/free block flag and a defect block flag stored in a used flag location 754 and a defect flag location 756 respectively of the flag field 752. Block level flags provide information concerning the

status of a whole block 727 of the memory bank and therefore, only a single block level flag entry is required in the flag locations 754 and 756 per block. The used/new block flag indicates whether the corresponding block 727 is currently being "used" to store information or whether it is available (or free) to store information. The defect block flag indicates whether the corresponding block 727 is defective.

In another embodiment, controller 510 (Fig. 6) monitors the status of each memory row location 728 of the memory bank using flags including a used/free row flag stored in the used flag location 754, a defect row flag stored in the defect flag location 756, an old row flag stored in an old flag location 758 of the flag field 752, an even sector move flag stored in an even sector move flag location 760, and an odd sector move flag stored in an odd sector move flag location 762. In this embodiment, the used/new flag indicates whether the corresponding memory row location 728 is currently being "used" to store information or whether it is available (or free) to store information. The defect flag indicates whether the memory block 727 is defective. If either of a corresponding pair of non-volatile memory locations 732, 733 is determined to be defective, then the whole memory block 727 is declared to be defective as indicated by the value in the defect flag location 756 being set, and the defective block can no longer be used. In a preferred embodiment, locations 758, 754, and 756 are included in a single 3-bit flag location 764.

The even and odd sector move flag locations 760, 762 store values indicating whether the corresponding even and odd sectors stored in the non-volatile memory sector location have been moved to another location within the non-volatile memory bank 506 (Fig. 6). For example, if an even sector of information stored collectively in a particular pair of even sector fields 734 , 742 of a row location 728 has been moved to another pair of even sector fields in the non-volatile memory bank 506, the value in the corresponding even sector move flag location 760 is set. Similarly, if an odd sector of information stored collectively in the odd sector fields 738, 746 of the same row location has been moved to another pair of odd sector fields in the non-volatile memory bank, then the value in the corresponding odd sector move flag location 672 is set. The location within the non-volatile memory bank 506 to which a sector of information has been moved is indicated in the LBA-PBA map stored in the SPM RAM 720 in an MVPBA address location, as taught in a patent application, filed by the inventors of this application, entitled "Moving Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", Serial No. 08/831,266, filed March 31, 1997, the

disclosure of which is incorporated herein by reference. In a preferred embodiment, locations 760 and 762 are formed by a single 2-bit move-flag location 766.

Fig. 8A shows a table diagram generally illustrating organization of an exemplary LBA-PBA map at 800, which is stored in SPM RAM 720 (Fig. 6), for translating a modified version of the host-provided LBA's to PBA's. The modified host-provided LBA is derived by dividing the host-provided LBA by the number of sectors with a block, as explained in more detail below. The depicted LBA-PBA map includes: a plurality of map row locations 802 which are addressable by a modified host-provided LBA or by a virtual PBA; a virtual PBA field 804 for storing a virtual PBA value identifying a block 727 (Fig. 7) within the memory bank; and a flag field 806 for storing flag information. As previously mentioned, the actual PBA specifies the location of a sector of information in the memory bank and the virtual PBA specifies the location of a block 727 (Fig. 7) in the memory bank. Virtual PBA values are retrieved by space manager 636 (Fig. 7) from the depicted map and transferred to port 648 of the flash state machine 642 for use in addressing blocks within memory bank 506.

Fig. 8B shows a block diagram illustrating a host-provided-LBA format 810 and an actual PBA format 820. LBA format 810 includes "offset bits" 812, which comprise the least significant bits of the host-provided LBA value. As explained above, in the preferred embodiment, each block 727 (Fig. 7) includes memory space for storing 32 sectors of information, each sector includes 512 bytes of user data and 16 bytes of overhead information. Because each block 727 (Fig. 7) includes 32 sectors in the preferred embodiment, five offset bits 812 are required to identify each of the 32 sectors in each block. In this embodiment, the translation of the host-provided-LBA to actual and virtual PBA values is performed by first masking the five least significant "offset" bits 812, of the host-provided-LBA, shifting the result to the right by 5 bits and using the shifted value as a modified host-provided LBA value or an "LBA-map-value" to address a map row location 802 in the LBA-PBA map 800 (Fig. 8A). This, in effect, is dividing the host-provided LBA by 32. The actual PBA value 820, which specifies the location of a sector within a block of the memory bank, is formed by concatenating offset bits 812 of the LBA value with a virtual PBA 822 value stored in the corresponding field 804 (Fig. 8A) of the LBA-PBA map. That is, the virtual PBA value 822 is used to identify a block within the memory bank and the five remaining offset bits 812 are used to address a sector within the identified block.

Upon initialization of memory system 600 (Fig. 6), the virtual PBA value stored in the virtual PBA field 804 of each map row location 802 is set to an all '1's state. Each time a block 727 (Fig. 7) is accessed by the controller, such as during a write operation, the virtual PBA value stored in the corresponding virtual PBA field 804 of the corresponding map row location is modified by the space manager controller 724 (Fig. 6) to specify a new virtual PBA value. When a block within the memory bank 506 is erased, the old virtual PBA value (the virtual PBA value corresponding to the erased block), rather than a modified version of the host-provided LBA, is used to address the SPM RAM 720 (Fig. 6) and the used flag, stored within the flag field of the SPM RAM 720, is cleared. This same 'used' flag within the flag field of the SPM RAM 720 is set at the time when the corresponding virtual PBA is updated pointing to the new block in the memory bank where sector information is maintained (step 1214).

Fig. 9 shows a timing diagram illustrating the timing of control, address, and data signals for a write operation performed by memory system 600 (Fig. 6) wherein two sectors of information are simultaneously written in the non-volatile memory bank 506 (Fig. 6) during a single write operation. The diagram includes: a wave form 902 representing a first flash signal which transmits time multiplexed command, address, and data information from flash state machine 642 (Fig. 6) of the controller via bus 680 (Fig. 6) to port 682 of the first flash memory chip; a wave form 904 representing a second flash signal which transmits time multiplexed command, address, and data signals from the flash state machine via bus 684 (Fig. 6) to port 686 of the second flash memory chip; a time line 905; and a plurality of control signal wave forms.

The control signal wave forms include: a wave form 906 representing a command line enable signal (CLE signal) transmitted from flash state machine 642 (Fig. 6) to the first and second flash memory chips via control bus 690 (Fig. 6); a wave form 908 representing an address line enable signal (ALE signal) transmitted from the flash state machine to the flash memory chips via the control bus; a wave form 910 representing a write enable signal (WE signal) transmitted from the flash state machine to the flash memory chips via the control bus; a wave form 912 representing a read enable signal (RE signal) transmitted from the flash state machine to the memory chips via the control bus; a wave form 914 representing a flash chip enable signal (FCE* signal) transmitted from chip enable signal output 700 (Fig. 6) of the flash state machine via chip enable line 698 to the first and second flash memory chips; a

wave form 916 representing a flash ready/busy signal (FRDY_BSY* signal) transmitted from outputs 708 and 710 (Fig. 6) of the first and second flash memory chips to the flash state machine via flash ready/busy signal line 706.

The write operation commences at a time t0 at which the FCE* signal (wave form 914) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to begin receiving command, address, data, and control signals. Prior to time t0, the FRDY_BSY* signal (wave form 916), transmitted from the flash memory chips to input 712 of the flash state machine (Fig. 6), is already activated indicating that the first and second flash memory chips are ready to receive access commands. At a subsequent time t1, the CLE signal (wave form 906) is activated, transitioning from a LOW state to a HIGH state, thereby enabling the first and second flash memory chips to read command signals. At a time t2, the first and second flash signals (wave forms 902 and 904) simultaneously transmit a serial data shift-in command signal 80H to the first and second flash memory chips via the first and second first split buses 680 and 684 respectively. At a time t3, while the serial data shift-in command signals 80H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the serial data command signals 80H. At a time t4, the CLE signal (wave form 906) is deactivated, transitioning back to the LOW state, thereby disabling the flash memory chips from reading command signals.

Also at time t4, the ALE signal (wave form 908) is activated, transitioning from a LOW state to a HIGH state, thereby enabling the first and second flash memory chips to read packets of address information. At times t5, t6, and t7, the first and second flash signals (wave forms 902 and 904) each transmit first, second, and third address packets ADD0, ADD1, and ADD2 respectively to the first and second flash memory chips. At a time t8, the ALE signal (wave form 908) is deactivated, transitioning from the HIGH state to a LOW state, thereby disabling the first and second flash memory chips from reading address information. During time intervals between times t5 and t6, t6 and t7, and t7 and t8, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the read the first, second, and third address packets ADD0, ADD1, and ADD2 respectively. The three address packets ADD0, ADD1, and ADD2 specify a row-portion 732, 733 within a first sub-block 730 (Fig. 16).

At a time t9, the first and second flash signals (wave forms 902 and 904) begin simultaneously transmitting interleaved even and odd data bytes wherein the even and odd bytes form one sector of information. The even bytes are transmitted to the first flash memory chip via bus 680 (Fig. 6) and the odd sector bytes are transmitted to the second flash memory chip via bus 684 (Fig. 6). The even data bytes D0, D2, D4,....D510 of the even sector are received by the first flash chip and stored in the first even sector field 734 (Fig. 16) of the corresponding location 732 of the first flash memory chip. This is done by storing a byte each time the write enable signal WE* (Wave form 910) is activated. The odd data bytes D1, D3, D5,....D511 of the even sector are received by the second flash chip and stored in the second even sector field 742 (Fig. 16) of the corresponding location 733 thereof with each byte being stored when the WE* signal is activated. At a time t10, the first and second flash signals (wave forms 902 and 904) complete transmission of the interleaved even and odd data bytes of the even sector.

Immediately after time t10, during an interval between time t10 and a time t11, the first flash signal (wave form 902) transmits four packets of filler information (FFH, hexadecimal F, equivalent binary value "1111," decimal value "15") to the first flash memory chip via the first split bus 680 (Fig. 6) while the second flash signal (wave form 904) transmits error correction codes (ECC) to the second flash memory chip via the second split bus 684 (Fig. 6). The filler information FFH transmitted during this time period is received by the first flash memory chip and stored in the first spare field 736 (Fig. 16). The error correction code transmitted during this time period is received by the second flash memory chip and stored in the first error correction field 744 (Fig. 16) of the nonvolatile memory section 733 of the second flash memory chip. This error correction code, generated by ECC logic unit 660 (Fig. 16), relates to the even sector transmitted during the preceding time interval between time t10 and t11.

At a time t11, the first and second flash signals (wave forms 902 and 904) begin simultaneously transmitting interleaved even and odd data bytes, synchronous with the write enable signal WE* (wave form 910), of an odd sector to the first and second flash memory chips via the first and second first split buses 680 and 684 (Fig. 6) respectively. The even data bytes D0, D2, D4,....D510 of the odd sector are received by the first flash chip and stored to the first odd sector field 738 (Fig. 16) of the corresponding location 732 of the first flash memory chip. The odd data bytes D1, D3, D5,....D511 of the odd sector are received by the

second flash memory chip and stored to the second odd sector field 746 (Fig. 16) of the corresponding location 733 of the second flash memory chip. At a time t12, the first and second flash signals (wave forms 902 and 904) complete transmission of the interleaved even and odd data bytes of the odd sector.

Immediately after time t12, during an interval between time t12 and a time t13, the first flash signal (wave form 902) transmits no information to the first flash memory chip thereby maintaining the value in corresponding storage location bytes of the first flash memory chip at FFH (hexadecimal) or all 1's in binary. Meanwhile, between time t12 and time t13, while the second flash signal (wave form 904) transmits error correction codes (ECC) to the second flash memory chip via the second split bus 684 (Fig. 6). The filler information FFH transmitted during this time period is received by the first flash memory chip and stored to the second spare field 740 (Fig. 16). The error correction code transmitted during this time period is received by the second flash memory chip and stored to the second error correction field 748 (Fig. 16) of the nonvolatile memory section 733 of the second flash memory chip. This error correction code, generated by ECC logic unit 660 (Fig. 16), relates to the odd sector transmitted during the preceding time interval between time t11 and t12.

At a time t17, the first and second flash signals (wave forms 902 and 904) each transmit a read command signal 70H to the first and second first and second flash memory chips via the first and second split buses 680 and 684 respectively. While the read command signals 70H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the read command signals 70H. At a time t18, the CLE signal (wave form 906) is deactivated, transitioning back to the LOW state, thereby disabling the flash memory chips from reading command signals.

At a time t18, the first and second flash signals (wave forms 902 and 904) each transmit a status command signal STATUS to the first and second first and second flash memory chips via the first and second split buses 680 and 684 respectively. While the read command signals 70H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the read command signals 70H.

Fig. 10 shows a table diagram generally illustrating the memory storage format, as depicted in Fig. 7, for storing a block of information in memory bank 506 (Fig. 6) wherein a

single sector is written to a particular memory row location of the memory bank. As shown, a memory row location 728 designated ROW1 has an even sector S2 and an odd sector S3 stored therein in accordance with the format described above in reference to Fig. 7. A memory row location 728 designated ROW2 has a single even sector S4 stored in the first and second even sector fields 734 and 742 of a corresponding pair of row-portions of the first and second flash memory chips 670, 672. Because no odd sector is required to be stored in this case, fields 736, 738, 746, 748, 750, and 752 are shown to be erased.

Fig. 11 shows a table diagram illustrating the alternative memory storage format, as depicted in Fig. 7, for storing a block of information in memory bank 506 (Fig. 6) wherein a single sector is written to a particular memory row location of the memory bank. As mentioned above, field 764 is a three bit field which is used for storing the old row flag in the first bit place, the used/free row flag in the second bit place, and the defect row flag in the third bit place. Also as described above, field 766 is a two bit field which is used for storing the even sector move flag in the first bit place and the odd sector move flag in the second bit place.

The memory row location designated ROW1, having sectors S2 and S4 stored therein, has a value "00" stored in field 766 indicating that both sectors have been moved elsewhere in the memory bank. The memory row location designated ROW2, having a single even sector S4 stored in the first and second even sector fields 734 and 742, has a value "01" stored in field 766 indicating that the information in S4 has been updated and now resides elsewhere in the memory bank. A value of logic state "0" generally indicates that moved sectors have been updated by the host. Therefore, when the remaining sectors are moved from the old block which was not updated by the host, it can be determined that these sectors are not to be overwritten by the old data during the move.

A memory location 728 designated ROW1 has an even sector S2 and an odd sector S3 stored therein in accordance with the format described above in reference to Fig. 7. A memory location 728 designated ROW2 has a single even sector S4 stored in the first and second even sector fields 734 and 742 of a corresponding pair of row-portions of the first and second flash memory chips 670, 672. Because no odd sector is required to be stored in this case, fields 736, 738, 746, 748, 750, and 752 are shown to be erased.

Fig. 12 is a flowchart illustrating a process of simultaneously writing two sectors of information to two memory units during a single write operation in accordance with the

present invention. In step 1202, the memory controller 510 (Fig. 6) receives host addressing information from host 504 which specifies addresses for one or more sector locations, in the form of a logical block address (host-provided LBA) or in the form of cylinder head sector (CHS) information. If the host addressing information is in the form of CHS information, the controller translates the CHS information to LBA information. As mentioned, the sectors are organized in blocks and therefore, the host-provided LBA's may correspond to sectors of more than one block. This information is used by microprocessor 620 (Fig. 6) as will be further discussed below.

Microprocessor 620 (Fig. 6) executes instructions, which are stored in code storage unit 626 (Fig. 6) to carry out the depicted process. In step 1204, a sector count value is set equal to the number of sector locations of a current block, being addressed by the host wherein a sector location may, for example, be comprised of fields 734 and 742 (Fig. 7) or fields 738 and 746 (Fig. 7) of the memory bank. The microprocessor determines at 1206 whether or not each of the sector locations specified by the host-provided LBA values has been accessed by the host before. This determination is made by reading the contents of the corresponding virtual PBA field 804 (Fig. 8A) of the LBA-PBA map 800 stored in SPM RAM 720 (Fig. 6). As explained above in reference to Fig. 8A, if the virtual PBA value corresponding to a host-provided LBA is set to the all '1's state, then the corresponding LBA was not accessed by the host before. Memory space in memory bank 506 is erased a block at a time. If any sectors of a block have been accessed since a last erasure of the block, then the block is indicated as having been accessed by virtue of the virtual PBA value in field 804 (Fig. 8A) of the corresponding map row location of the LBA-PBA map being a value other than "all 1's".

If it is determined that one or more sector locations, of the current block, specified by the host-provided-LBA's have been accessed previously by the host, the write process proceeds to step 1210 in which microprocessor 620 (Fig. 6) sets the corresponding one of the move flags 760, 762 (Fig. 7) corresponding to the current sector location, and the write process proceeds to step 1208. As earlier discussed, maintaining the 'move' flag in non-volatile memory is optional and may be entirely eliminated without departing from the scope and spirit of the present invention. In the absence of move flags, the microprocessor maintains the status of sectors as to whether or not they have been moved to other blocks. This is done by keeping track of two values for each block. One value is the starting sector

location within a block where sectors have been moved and the second value is the number sectors within the block that have been moved. With these two values, status information as to whether or not and which sectors of a block have been moved to other block(s) may be reconstructed.

If it is determined, at step 1206, that none of the sector locations of the current block specified by the host-provided-LBA have been previously accessed, the write process proceeds directly to step 1208.

In step 1208, the space manager 636 (Fig. 6) of the controller searches for a free (or unused) block, such as block 727 (Fig. 7) located within the nonvolatile memory bank, each free block being identified by a specific virtual PBA value. The microprocessor determines at 1212 whether a free block is located, and if not, an error is reported by the controller 510 (Fig. 6) to the host indicating that the nonvolatile memory bank is unable to accommodate further storage of information. As this can result in a fatal system error, the inventors of the present invention have exercised great care in preventing this situation from occurring.

Once a free block within the nonvolatile memory is located at step 1208, the depicted process proceeds to step 1214. In step 1214, microprocessor 620 prompts space manager 636 (Fig. 6) to assign a virtual PBA value 822 (Fig. 8B) to the free block found in step 1208. This virtual PBA value is stored in the LBA-PBA map 800 (Fig. 8A) in a map row location 802 (Fig. 8A) identified by the masked bits 814 (Fig. 8B) of the host-provided LBA corresponding to the current block. The masked bits 814 (Fig. 8B) of the current host-provided LBA are obtained by shifting the host-provided LBA to the right by the 5 offset bits (or by dividing by 32). For example, if the host-identified LBA is 16H (hexadecimal notation), the row in which the virtual PBA is stored is row 0. Also at step 1214, the microprocessor appends the 'offset' bits 812 (Fig. 8B) to the virtual PBA corresponding to the found free block to obtain an actual PBA value 820 (Fig. 8B). At 1216, the microprocessor determines whether the actual PBA value is an even or odd value. At 1216, alternatively, the host-provided LBA may be checked in place of the actual PBA value to determine whether this value is odd or even.

If it is determined at 1216 that the actual PBA value is even, the process proceeds to 1218 at which the microprocessor determines whether the sector count is greater than one, i.e., there is more than one sector of information to be written at the point the controller requests that more than one sector to be transferred from the host to the internal buffer of the

controller and the process proceeds to 1232 at which the microprocessor determines whether two sectors of information have been transferred from the host to the data buffer 614 (Fig. 6) (through the host interface circuit 610). That is, where there is more than one sector of information that needs to be written to nonvolatile memory, as detected by the flash state machine 642, two sectors of information are transferred at-a-time from the host to the data buffer 614. The data buffer 614 is used to temporarily store the sectors' information until the same is stored into the memory bank 506. In the preferred embodiment, each sector includes 512 bytes of user data and 16 bytes of overhead information.

Where two sectors of information have not yet been transferred to the data buffer 614, the microprocessor waits until such a transfer is completed, as shown by the 'NO' branch loop at 1232.

At step 1234, the microprocessor initiates the writing of the two sectors that have been temporarily saved to the data buffer to the memory bank 506 (Fig. 6) by issuing a write command, followed by address and data information. The write operation at step 1234 is performed according to the method and apparatus discussed above relative to Figs. 7 and 9.

Upon completion of writing two sectors of information, the write operation is verified at 1235. If information was not correctly programmed into the sectors at step 1234, the process continues to step 1237 where a defect management routine is performed, as will be discussed in greater detail below. After execution of the defect management routine, the sector count is decremented by two at step 1236. At 1235, if the write operation was verified as being successful, step 1236 is executed and no defect management is necessary. The microprocessor then determines at 1238 whether the sector count is equal to zero and if so, it is assumed that no more sectors remain to be written and the process proceeds to 1228. If, however, more sectors need to be written the process proceeds to step 1240 at which the host-provided LBA is incremented by two to point to the next sector that is to be written.

At step 1240, the microprocessor determines whether the last sector of the block has been reached. The block boundary is determined by comparing the 'offset' value of the current LBA to the number of sectors in a block, and if those values are equal, a block boundary is reached. For example, in the preferred embodiment, since a block includes 32 sectors, the 'offset' value of the current LBA is compared against '32' (in decimal notation). If alternatively, a block is defined to have other than 32 sectors, such as 16 sectors, the latter is compared against the 'offset'. If a block boundary in the nonvolatile memory is reached,

the write process continues from step 1206 where the virtual PBA value corresponding to the current LBA value is checked for an all '1's condition and so on. If a block boundary is not reached at step 1242, the write process continues from step 1218.

At step 1218, if it is determined that the sector count is not greater than one, the microprocessor proceeds to determine at 1220 whether data buffer 614 (Fig. 6) has received at least one sector of information from the host. If not, the microprocessor waits until one sector of information is transferred from the host to the data buffer 614. Upon receipt of one sector of information, writing of the next sector is initiated and performed at step 1222 according to the method and apparatus discussed above relative to Figs. 10 and 11. Upon completion of writing a sector of information, the write operation is verified at 1223. If information was not correctly programmed into the sector at step 1222, the process continues to step 1225 where a defect management routine is performed, as will be discussed in greater detail below. After execution of the defect management routine, at step 1224, the sector count is decremented by one. If at 1223, it is determined that the write operation was correctly performed, the process continues to step 1224 and no defect management routine is executed. At 1226, the microprocessor determines whether the sector count is equal to zero and, if not, the host-provided LBA is incremented by one and the write process continues to step 1242 where the microprocessor checks for a block boundary as explained above.

If at step 1226, as in step 1238, it is determined that no more sectors remain to be written, i.e. the sector count is zero, the depicted process proceeds to 1228 at which the microprocessor determines whether the move flag is set. As noted above, the move flag would be set at step 1210 if it was determined at 1206 that an LBA was being re-accessed by the host.

If it is determined at 1228 that the move flag is not set, the write process ends. However, upon a determined at 1228 that the move flag is set, the block is updated. That is, those sectors of the current block that were not accessed are moved to corresponding sector locations in the block within memory bank 506 identified by the virtual PBA value assigned in step 1214 to the free block found in step 1208. This is perhaps best understood by an example.

Let us assume for the purpose of discussion that the sectors identified by LBAs 1, 2, 3, 4, 5 and 6 have already been written and that the host now commands the controller to write data to sectors identified by LBAs 3, 4 and 5. Further, let us assume that during the first write

process when LBAs 1-6 were written, they were stored in a block location in the memory bank 506 (Fig. 6) identified by a virtual PBA value of "3" and the LBA locations 3, 4 and 5 are now (during the second write process) being written to a location in the memory bank identified by a virtual PBA value of "8". During writing of locations identified by host-provided LBA values of 3, 4, and 5, the microprocessor at step 1206 determines that these block locations are being re-accessed and the move flag at 1210 is set. Furthermore, at step 1230, after the sectors, identified by host-provided LBAs 3, 4, and 5, have been written to corresponding sectors of the block identified by virtual PBA "8", sectors in the block identified by virtual PBA "3" that were not re-accessed during the write operation are moved from the block identified by virtual PBA "3" to corresponding sector locations of the block identified by virtual PBA "8" and the block identified by virtual PBA "3" is thereafter erased. This example assumes that remaining sectors of the block identified by virtual PBA "3", such as sectors 0 and 7-31 (assuming there are 32 sectors in a block), were not accessed since the last erase of the block in which they reside and therefore contain no valid sector information. Otherwise, if those sectors were previously accessed, then they would also be moved to the virtual PBA location 8.

Step 1230 may be implemented in many ways. The inventors of the present invention disclose various methods and apparatus which may be alternatively employed for performing the move operation of step 1230. In patent applications, Serial No. 08/946,331 entitled "Moving Sequential Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", filed on Oct. 7, 1997, and Serial No. 08/831,266 entitled "Moving Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", filed on March 31, 1997, the disclosures of which are herein incorporated by reference.

Fig. 12a shows the steps performed by the micorprocessor if the defect management routine at steps 1237 and 1225 (in Fig. 12) is executed. The block management routine is executed when the write operation is not successfully verified; the block(s) being programmed is in some way defective and a different area in the nonvolatile memory, i.e. another block need be located for programming therein.

At step 1600, the block that was being unsuccessfully programmed is marked as "defective" by setting the "defect" flags 756 (in Fig. 7). At step 1602, the space manager within the controller is commanded to find a free block. At step 1604, the information that would have been programmed at steps 1234 and 1222 (in Fig. 12) i.e. the block marked

"defective" is programmed into corresponding sector locations within the free block found in step 1602.

At step 1606, the block marked "defective" is checked for the presence of any sector information that was previously written thereto successfully. If any such sectors exist, at step 1608, these previouly-programmed sectors are moved to the free block, as is additional block information in the process of Fig. 12.

Fig. 13 shows a table diagram generally illustrating a memory storage format for storing a block, including 32 sectors, of information in memory bank 506 in accordance with an alternative embodiment of the present invention. In this embodiment, an even sector is stored in a first row portion located in a first of the two memory units and an odd sector is stored in a second row portion located in the second of the two memory units. In the depicted embodiment, memory bank 506 includes a plurality of M blocks 1302 designated BLCK0, BLCK1, BLCK(M-1) each having a physical block addresses (PBA). Each of the blocks 1302 includes a plurality of N memory row locations 1304, and in a preferred embodiment, N=16. Each block 1302 of memory bank 506 is comprised of a first sub-block 1306 of first flash memory chip 670, and a corresponding second sub-block 1308 of second flash memory chip 672 wherein the corresponding sub-blocks are identified by the same virtual PBA. Each memory row location 1304 includes a first row-portion 1310 and a corresponding second row-portion 1312. In the depicted embodiment each of the first and second row-portions 1310, 1312 includes storage for 512 bytes of data information plus additional storage space for error correction information (ECC information) and flag information.

Each of the first row-portions 1310 includes an even sector field 1314 for storing an even sector (S0, S2, S4,...) of information, and an even sector error correction field 1316 for storing error correction information corresponding to the even sector stored in field 1314. Each of the second row-portions 1312 includes an odd sector field 1318 for storing an odd sector (S1, S3, S5,...) of information, an odd sector error correction field 1320 for storing error correction information corresponding to the odd sector stored in 1318, a block address field 1322, and a flag field 1324. It is understood in the present invention that field 1314 could alternatively be used to store an odd sector while field 1318 could alternatively be used to store an even sector. Also, first row-portion 1310 could alternatively be used for storing the block address and flags.

Flag field 1324 is used for storing flag information which is used by controller 510 (Fig. 6) during access operations as further explained below. Block address field 1322 is used for storing the block address permanently assigned to block 1302, such as "0" for BLCK0. Only a single block address entry is required in the block address field per block. In a preferred embodiment, a block address entry is entered in block address field 1322 of the last row 1304, which is row 15.

In this alternative embodiment, the first and second split buses 680, 684 (Fig. 6) include lines coupled to receive data bytes of the even and odd sectors respectively. The controller 510 (Fig. 6) writes two sectors simultaneously by simultaneously writing a byte of an even sector and an odd sector simultaneously via the first and second split buses 680, 684 (Fig. 6), respectively. The split buses 680, 684 (Fig. 6) also provide for: transmission of ECC information between the flash memory chips and the flash state machine 642 and ECC logic unit 660 of the memory controller 510; and transmission of address information from flash state machine 642 to the flash memory chips.

Fig. 14 shows a timing diagram illustrating the timing of control signals, address signals, and data signals for an erase operation of the memory system of Fig. 6. The diagram includes: the wave form 902 representing the first flash signal which transmits time multiplexed command, address, and data information from the flash state machine 642 (Fig. 6) via first split bus 680 (Fig. 6) to the first flash memory chip; the wave form 904 representing the second flash signal which transmits time multiplexed command, address, and data signals transmitted from the flash state machine via second split bus 684 (Fig. 6) to the second flash memory chip; a time line 1450; and a plurality of control signal wave forms. The control signal wave forms, all of which are described above, include: wave form 906 representing the command line enable (CLE) signal; wave form 908 representing the address line enable (ALE) signal; wave form 910 representing the write enable (WE) signal; wave form 912 representing the read enable (RE) signal; wave form 914 representing the flash chip enable (FCE*) signal; and wave form 916 representing the flash ready/busy signal (FRDY_BSY* signal).

The erase operation commences at a time E0 at which the FCE* signal (wave form 914) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to begin receiving command, address, and data signals. At a subsequent time E1, the CLE signal (wave form 906) is activated, transitioning from a LOW state to a HIGH

state, thereby enabling the first and second flash memory chips to read command signals. At a time E2, the first and second flash signals (wave forms 902 and 904) each transmit a command signal. The first flash signal (wave form 902) transmits an 'erase set' command, 60H, via the first split bus 680 (Fig. 6) to the first flash memory chip while the second flash signal (wave form 904) transmits a read status command signal 70H via the second split bus 684 (Fig. 6) to the second flash memory chip. At a time E3, while the command signals 60H and 70H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the command signals 60H and 70H. At a time E4, the CLE signal (wave form 906) is deactivated, transitioning back to the LOW state, thereby disabling the flash memory chips from reading command signals.

Also at time E4, the ALE signal (wave form 908) is activated, transitioning from a LOW state to a HIGH state, thereby enabling the first and second flash memory chips to read packets of address information. At times E5 and E6, the first flash signal (wave form 902) transmits first and second address packets ADD0 and ADD1 respectively to the first flash memory chip wherein the first and second address packets ADD0 and ADD1 specify a sub-block 730 (Fig. 7) of the first flash memory chip 670 of the memory bank. At a time E7, the ALE signal (wave form 908) is deactivated. During time intervals between times E3 and E4, and E4 and E5, the WE signal (wave form 910) transitions to the LOW state to enable the flash memory chip to read the address packets.

At a time E8, the CLE signal (wave form 906) is again activated to enable the first and second memory chips to read command signals. At a time E9, the first flash signal (wave form 902) transmits DOH, which is an 'erase confirm command' to the first flash memory chip. This command, as sampled by the CLE signal, actually initiates the erase operation within the flash chips, after which, the contents of data fields 734 and 738 of each memory row portion 732 of the addressed sub-block 730 (Fig. 7) of the first flash memory chip 670 are erased, i.e. set to an "all 1's" state. At a time E10, the FRDY-BSY* signal (wave form 912) transitions from a HIGH state to a LOW state to indicate to the flash state machine 642 (Fig. 6) that at least one of the flash memory chips is busy.

At a time E11, the CLE signal (wave form 906) is activated to enable the first and second flash memory chips to read command signals. At a time E12, the first and second flash signals (wave forms 902 and 904) each transmit a command signal. The first flash

signal (wave form 902) transmits a read command signal 70H via the first split bus 680 (Fig. 6) to the first flash memory chip while the second flash signal (wave form 904) transmits an erase command signal 60H via the second split bus 684 (Fig. 6) to the second flash memory chip. At a time E13, while the command signals 70H and 60H are active, the WE signal (wave form 910) transitions to the LOW state to enable the first and second flash memory chips to read the command signals 60H and 70H. At a time E14, the CLE signal (wave form 906) is deactivated to disable the flash memory chips from reading command signals and the ALE signal (wave form 908) is activated thereby enabling the first and second flash memory chips to read packets of address information. At times E15 and E16, the second flash signal (wave form 904) transmits first and second address packets ADD0 and ADD1 respectively to the second flash memory chip wherein the first and second address packets ADD0 and ADD1 specify a sub-block 731 (Fig. 7) of the second flash memory chip 672 of the memory bank. At a time E17, the ALE signal (wave form 908) is deactivated. During time intervals between times E13 and E14, and E14 and E15, the WE signal (wave form 910) enables the flash memory chips to read the address packets. At a time E18, the CLE signal (wave form 906) is again activated to enable the first and second memory chips to read command signals. At a time E19, the first flash signal (wave form 902) transmits DOH to the first flash memory chip to erase the contents of data fields 734 and 738 of each memory row portion 732 of the specified block and thereby set them to an "all 1's" state.

To summarize, during a time interval TEB1, between the times E0 and E11, the memory controller erases an addressed sub-block 730 (Fig. 7) of the first flash memory chip 670. Also, during a time interval TEB2, between the times E11 and E20, the memory controller erases a corresponding addressed sub-block 731 (Fig. 7) of the second flash memory chip 672. At a time E21, the FRDY_BSY* signal (wave form 916) transitions from a LOW state to a HIGH state to indicate to the flash state machine 642 (Fig. 6) that both of the flash memory chips are finished with the erase operation.

Immediately after time E21, the first and second flash signals (wave forms 902 and 904) each transmit a read status command signal 70H to the first and second flash memory chips respectively. While the read command signals 70H are active, the WE signal (wave form 910) transitions to the LOW state thereby enabling the first and second flash memory chips to read the read command signals 70H. At a time E22, the first and second flash signals (wave forms 902 and 904) both transmit a status data back to the controller.

So, the status of both flash memory chips are read simultaneously after the erase operation is performed on the two corresponding addressed sub-blocks of the flash memory chips as described above.

If either of the sub-blocks 730, 731 of the memory chips has an error, the entire block 727 (Fig. 7) within the chips is marked defective by setting the contents of the defect flag 756 (Fig. 7) in the second flash memory chip 672.

Fig. 15 is a flowchart illustrating a process of erasing a block, including a first sub-block stored in a first memory unit and a second sub-block stored in a second memory unit, in accordance with the present invention. Microprocessor 620 (Fig. 6) executes instructions, which are stored in code RAM 626 (Fig. 6) to carry out the depicted process.

In step 1502, microprocessor 620 (Fig. 6) loads a block address to be erased. In step 1504, the microprocessor initiates the erase operations described above in reference to the timing diagram at 1400 (Fig. 14). At 1506, the microprocessor determines whether the erase operation is finished by reading the flash ready/busy (FRDY_BSY*) signal (wave form 916 of Fig. 14) which transitions from a LOW state to a HIGH state to indicate to the flash state machine 642 (Fig. 6) that both of the flash memory chips are finished with the erase operation. At 1508, the microprocessor reads the status of the flash chips 670, 672 (Fig. 6). At 1508, the microprocessor determines whether the erase operation performed in step 1504 was successful in both of the flash chips 670, 672 (Fig. 6) and, if so, the process ends. If it is determined that the erase operation performed in step 1504 was not successful in both of the flash chips, then the microprocessor marks the block in both of the flash chips 670, 672 defective.

Although the present invention has been described in terms of specific embodiments, it is anticipated that alterations and modifications thereof will no doubt become apparent to those skilled in the art. It is therefore intended that the following claims be interpreted as covering all such alterations and modification as fall within the true spirit and scope of the invention.

What is claimed is:

CLAIMS

1. A memory storage device for storing information organized in sectors within a nonvolatile memory bank, each said sector including a user data portion and an overhead portion, said sectors being organized into blocks, each sector identified by a host-provided logical block address (LBA) and an actual physical block address (PBA) derived from a virtual PBA, each block being identified by a modified LBA derived from said host-provided LBA and said virtual PBA, said host-provided LBA being received by said storage device from the host for identifying a sector of information to be accessed, said actual PBA developed by said storage device for identifying a free location within said memory bank wherein said accessed sector is to be stored, said storage device comprising:

a memory controller coupled to said host; and

a nonvolatile memory bank coupled to said memory controller via a memory bus, said memory bank including a first non-volatile semiconductor memory unit and a second non-volatile semiconductor memory unit, said memory bank having storage blocks each of which includes at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, each said memory row location providing storage space for two of said sectors,

wherein said memory controller accesses two sectors of information simultaneously.

2. A memory storage device as recited in claim 1 wherein said memory controller includes:

a data buffer for temporarily storing said sector-organized information;

a microprocessor;

a space manager including a space manager controller and a space manager memory unit for maintaining a map for translating said LBA to said PBA; and

an error correction code logic unit for performing error coding and correction operations on said sector-organized information.

3. A memory storage device as recited in claim 1 wherein:

each of said first row-portions includes,

a first even sector field for storing even data bytes of an even sector, and
a first odd sector field for storing even data bytes of an odd sector;
each of said second row-portions includes,
a second even sector field for storing odd data bytes of said even sector, and
a second odd sector field for storing odd data bytes of said odd sector;
said memory bus includes,
a first split bus coupled to transmit said even data bytes of said sectors between said memory controller and said first memory unit;
a second split bus coupled to transmit said odd data bytes of said sectors between said memory controller and said second memory unit.

4. A memory storage device as recited in claim 3 wherein each of said second row-portions further includes:
a first error correction field for storing error correction information corresponding to said even sector;
a second error correction field for storing error correction information corresponding to said odd sector;
a block address field for storing said PBA which specifies an address of said corresponding block, and
a flag field for storing information indicative of the status of said corresponding block.

5. A memory storage device as recited in claim 4 wherein each of said first row-portions further includes:
a first error correction field for storing error correction information corresponding to said even sector;
a second error correction field for storing error correction information corresponding to said odd sector; and
a block address field for storing said PBA which specifies an address of said corresponding block, and
a flag field for storing information indicative of the status of said corresponding block.

6. A memory storage device as recited in claim 5 wherein:

each of said first and second row-portions includes storage space for 512 bytes of said user data plus an additional 16 bytes of storage space for said overhead information; and

said first split bus and said second split bus each include 8 bit lines.

7. A memory storage device as recited in claim 6 wherein said controller includes:

means for accessing an even sector of information by simultaneously accessing said first and second even sector fields of corresponding row-portions of said first and second memory units via said first and second split buses; and

means for accessing an odd sector of information by simultaneously accessing said first and second odd sector fields of corresponding row-portions of said first and second memory units via said first and second split buses.

8. A memory storage device as recited in claim 7 wherein said controller includes error correction code logic for performing error coding and correction operations on said sector-organized information.

9. A memory storage device as recited in claim 8 wherein said flag field is used to store block level flags including:

a used/free block flag indicating whether said corresponding block is currently being used to store information; and

a defect block flag indicating whether said corresponding block is defective

10. A memory storage device as recited in claim 9 wherein each of said corresponding sub-blocks are identified by a single PBA value.

11. A memory storage device as recited in claim 10 wherein said non-volatile memory units are flash memory chips.

12. A memory storage device as recited in claim 11 wherein:

each of said first row-portions includes a first sector field for storing data bytes of a first sector;

each of said second row-portions includes a second sector field for storing data bytes of a second sector;

said memory bus includes,

a first split bus coupled to transmit (least significant ?) data bytes of said sectors between said memory controller and said first memory unit;

a second split bus coupled to transmit (most significant ?) data bytes of said sectors between said memory controller and said second memory unit.

13. A memory storage device as recited in claim 12 wherein:

means for simultaneously accessing said first and second sector fields of corresponding row-portions of said first and second memory units via said first and second split buses

14. In a storage device including a non-volatile memory bank and a controller coupled to said memory bank via a memory bus, said memory bank including a first non-volatile memory unit and a second non-volatile memory unit, said memory bank having storage locations defined by blocks, each block for storing a sectors of information having a user data portion and an overhead portion, each block having associated therewith a modified logical block address (LBA) derived from a host-provided LBA value and an actual physical block address (PBA) derived from a virtual PBA value, said host-provided LBA value received by said controller from the host for identifying a sector of information to be accessed, said actual PBA developed by said storage device for identifying a free location within said memory bank wherein a sector of information identified by the host is to be stored, each block including at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, a process of writing a first sector and a second sector to said memory bank during a single write operation, said process including the steps of:

simultaneously providing a write command to said first and second memory units;

addressing one of said memory row locations of said memory bank by simultaneously addressing corresponding first and second row portions of said addressed row location;

simultaneously providing a first data byte of said first and second sectors to said first memory unit, and a second data byte of said first and second sectors to said second memory unit.

15. In a storage device including a memory bank and a controller as recited in claim 14 wherein said first byte is an even data byte of one of said first and second sectors and said second data byte is odd data byte of one of said first and second sectors.

16. In a storage device including a memory bank and a controller as recited in claim 15 wherein said first data byte is a data byte of said first sector and said second data byte is a data byte of said second sector.

17. In a storage device including a memory bank and a controller as recited in claim 16 wherein an even sector and an odd sector are stored in a single memory row location and wherein even user data bytes of said first and second sectors are stored in said first memory unit and odd user data bytes of said first and second sectors are stored in said second memory unit and wherein overhead information associated with said first and second sectors is stored in one of said first and second row portions.

18. In a storage device including a memory bank and a controller as recited in claim 17 wherein a block address entry is entered in block address field in a last of said row locations of each said block.

19. In a storage device including a memory bank and a controller as recited in claim 18 wherein said controller accesses an even sector of information stored collectively in said first and second flash memory chips by simultaneously accessing first and second even sector fields of corresponding row-portions of said first and second flash memory chips via said first and second split buses

said first and second split buses including lines coupled to receive said even and odd data bytes respectively of a sector of information

said controller accesses an odd sector of information stored collectively in said first and second flash memory chips by simultaneously accessing said first and second odd sector fields , via said first and second split buses

said split buses also providing for transmission of ECC information between said flash memory chips and said flash state machine and ECC logic unit of said memory controller ; and transmission of address information from flash state machine to said flash memory chips.

20. In a storage device including a memory bank and a controller as recited in claim 19 wherein said flag field is used to store block level flags including:
a used/free block flag indicating whether said corresponding block is currently being used to store information
a defect block flag indicating whether said corresponding block is defective

21. In a storage device including a memory bank and a controller as recited in claim 20 wherein each of said corresponding sub-blocks are identified by a single virtual PBA value.

22. In a storage device including a nonvolatile memory bank and a controller coupled to said memory bank via a memory bus, said memory bank including a first non-volatile memory unit and a second non-volatile memory unit, said memory bank having storage blocks for storing information sectors each of which includes a user data portion and an overhead portion, each block having associated therewith a logical block address (LBA) and a physical block address (PBA), said LBA provided by a host to said controller for identifying a block to be accessed, said PBA developed by said storage device for identifying a free location within said memory bank wherein said accessed block is to be stored, each block including at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, a process of writing a first sector and a second sector to said memory bank during a single write operation, said process including the steps of:

simultaneously providing a write command to said first and second memory units;

addressing one of said memory row locations of said memory bank by simultaneously addressing corresponding first and second row portions of said addressed row location;

storing even data bytes of an even sector to a first even sector field of said first row-portion;

storing even data bytes of an odd sector to a first odd sector field of said first addressed row-portion;

storing odd data bytes of said even sector to a second even sector field of said second addressed row-portion; and

storing odd data bytes of said odd sector of information to a second odd sector field of said second addressed row-portion.

23. In a storage device including a memory bank and a controller as recited in claim 22 further including the steps of:

determining even sector error correction information corresponding to said even sector of information and odd sector error correction information corresponding to said odd sector of information;

storing said even sector error correction information to a first error correction field of said second addressed row-portion; and

storing said odd sector error correction information to a second error correction field of said second addressed row-portion.

24. In a storage device including a nonvolatile memory bank and a controller coupled to said memory bank via a memory bus, said memory bank including a first non-volatile memory unit and a second non-volatile memory unit, said memory bank having blocks for storing sectors of information each of which includes a user data portion and an overhead portion, each said block having associated therewith a logical block address (LBA) and a physical block address (PBA), said LBA provided by a host to said controller for identifying a sector of information to be accessed, said PBA developed by said storage device for identifying a free block location within said memory bank wherein said accessed block is to be stored, each block including at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, said controller including a data buffer; a process of writing sector-organized information to said memory bank, said process including the steps of:

receiving host-provided LBA values from said host, each said host-provided LBA value for identifying a sector of information;

modifying a current host-provided LBA to identify a block of sectors of information;

providing a map having map row locations identified by said modified LBA values or virtual PBA values, said map used as a look-up-table for storing virtual PBA values corresponding to modified LBA values;

setting a sector count value equal to the number of sectors of information identified by the host;

searching for a free block within said memory bank identified by a current virtual PBA;

storing said current virtual PBA, corresponding to said free block, in a map row location identified by said modified current LBA in said map;

determining whether said current host-provided LBA is even and whether said sector count is greater than one;

if said current actual PBA value is even and said sector count is greater than one,

simultaneously writing two sectors of information, one sector identified by said current host-provided LBA and a second sector of information identified by said current host-provided LBA plus one, said first sector being written to said first non-volatile memory unit and said second sector being written to second non-volatile memory units of said memory bank,

decrementing said sector count by two,

determining whether said sector count is equal to zero,

if said sector count is not equal to zero, increasing said current host-provided LBA value by two to point to the next sector that is to be written,

if said current host-provided LBA value or said sector count is not greater than one,

simultaneously writing even data bytes of a current sector of information identified by said current host-provided LBA to said first non-volatile memory unit and odd data bytes of said current sector to said second non-volatile memory unit,

decrementing said sector count by one,

determining whether said sector count is equal to zero, and

if said sector count is not equal to zero, increasing said current host-provided LBA by one to point to the next sector of information that is to be written.

25. In a storage device including a memory bank and a controller as recited in claim 24, wherein said step of simultaneously writing said two current sectors to said first and second non-volatile memory units of said memory bank, includes the steps of:

simultaneously providing a write command to said first and second memory units;

addressing one of said memory row locations of said memory bank by simultaneously addressing corresponding first and second row portions of said addressed row location;

storing even data bytes of an even sector to a first even sector field of said first addressed row-portion;

storing even data bytes of an odd sector to a first odd sector field of said first addressed row-portion;

storing odd data bytes of said even sector to a second even sector field of said second addressed row-portion; and

storing odd data bytes of said odd sector of information to a second odd sector field of said second addressed row-portion.

26. In a storage device including a memory bank and a controller as recited in claim 25, further including the steps of:

determining whether or not each of said addressed sector locations of a current block has been accessed since the last erasure thereof; (by reading the contents of the corresponding virtual PBA field of the LBA-PBA map);

setting a move flag corresponding to said current block if said current block has been accessed since the last erasure thereof;

subsequent to performing a write operation

determining whether said move flag is set;

if said move flag is set, updating said current block by moving those of said sectors not written to which belong to said current block to corresponding sector locations in said free block.

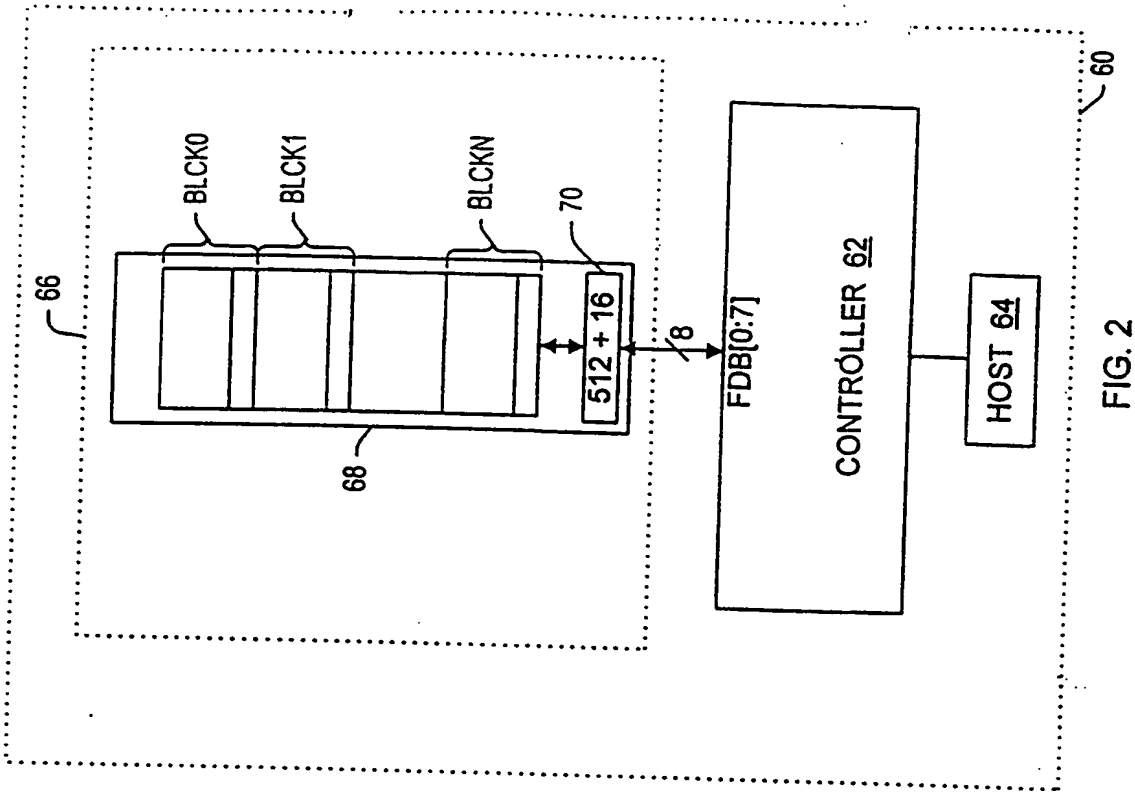


FIG. 2
(PRIOR ART)

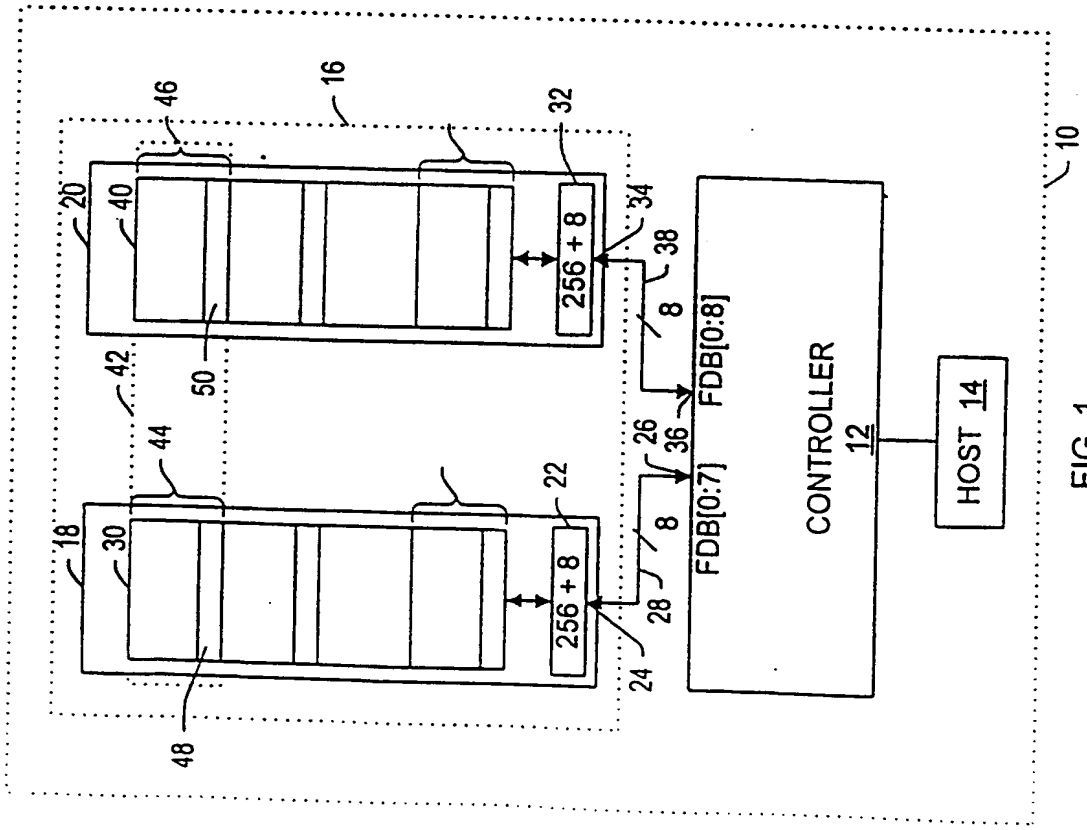


FIG. 1
(PRIOR ART)

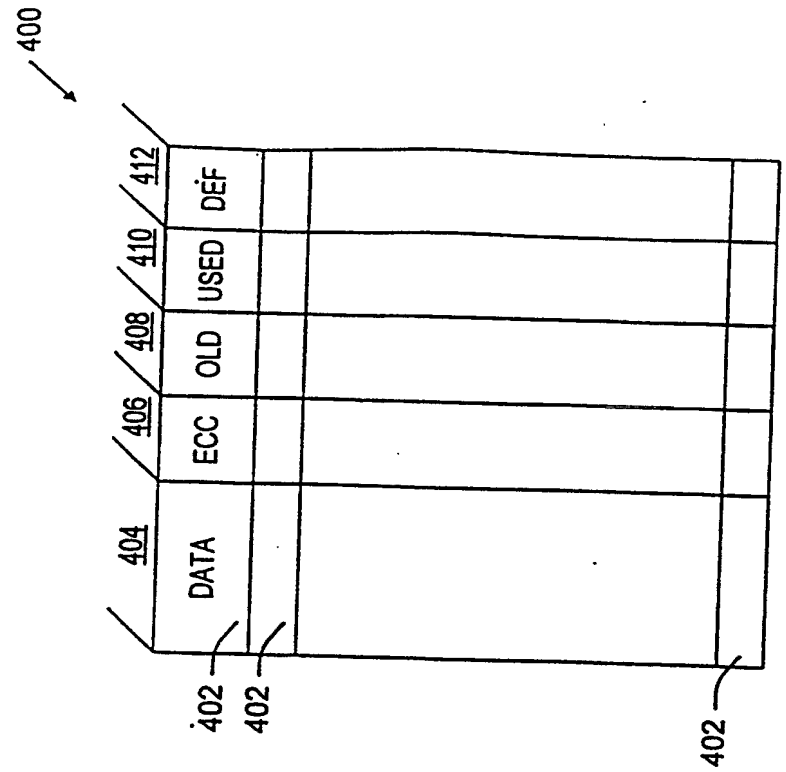


FIG. 4
(PRIOR ART)

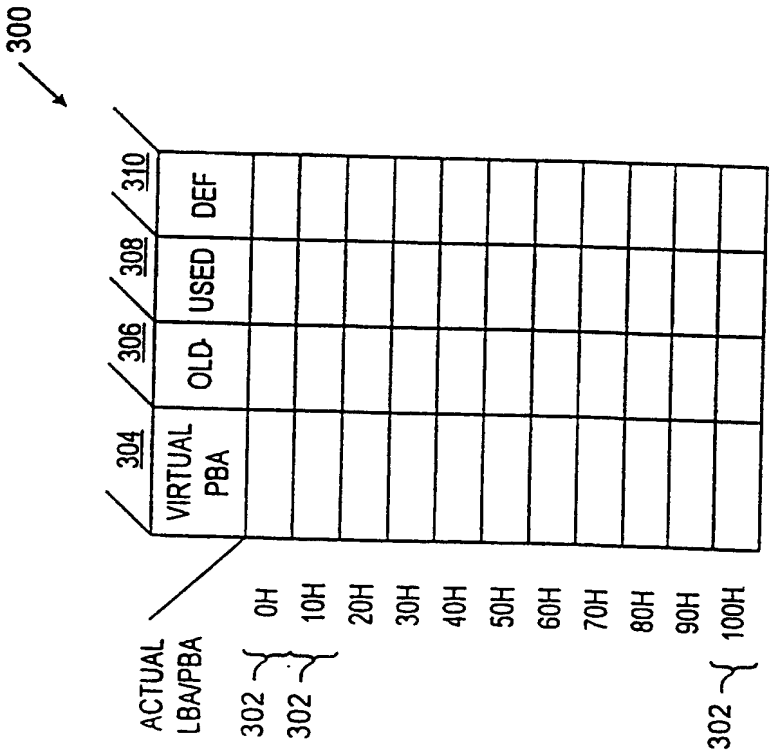


FIG. 3
(PRIOR ART)

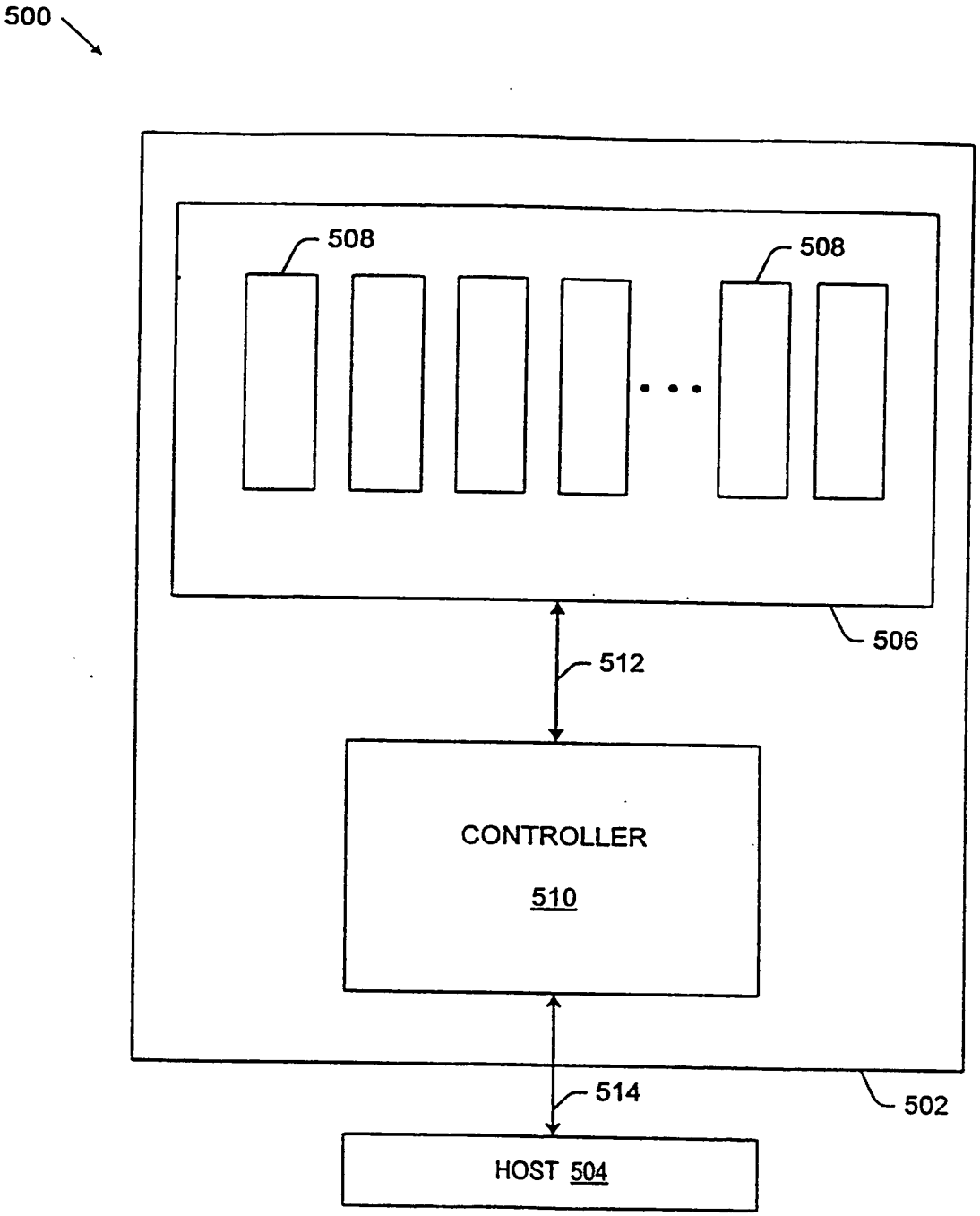


FIG. 5

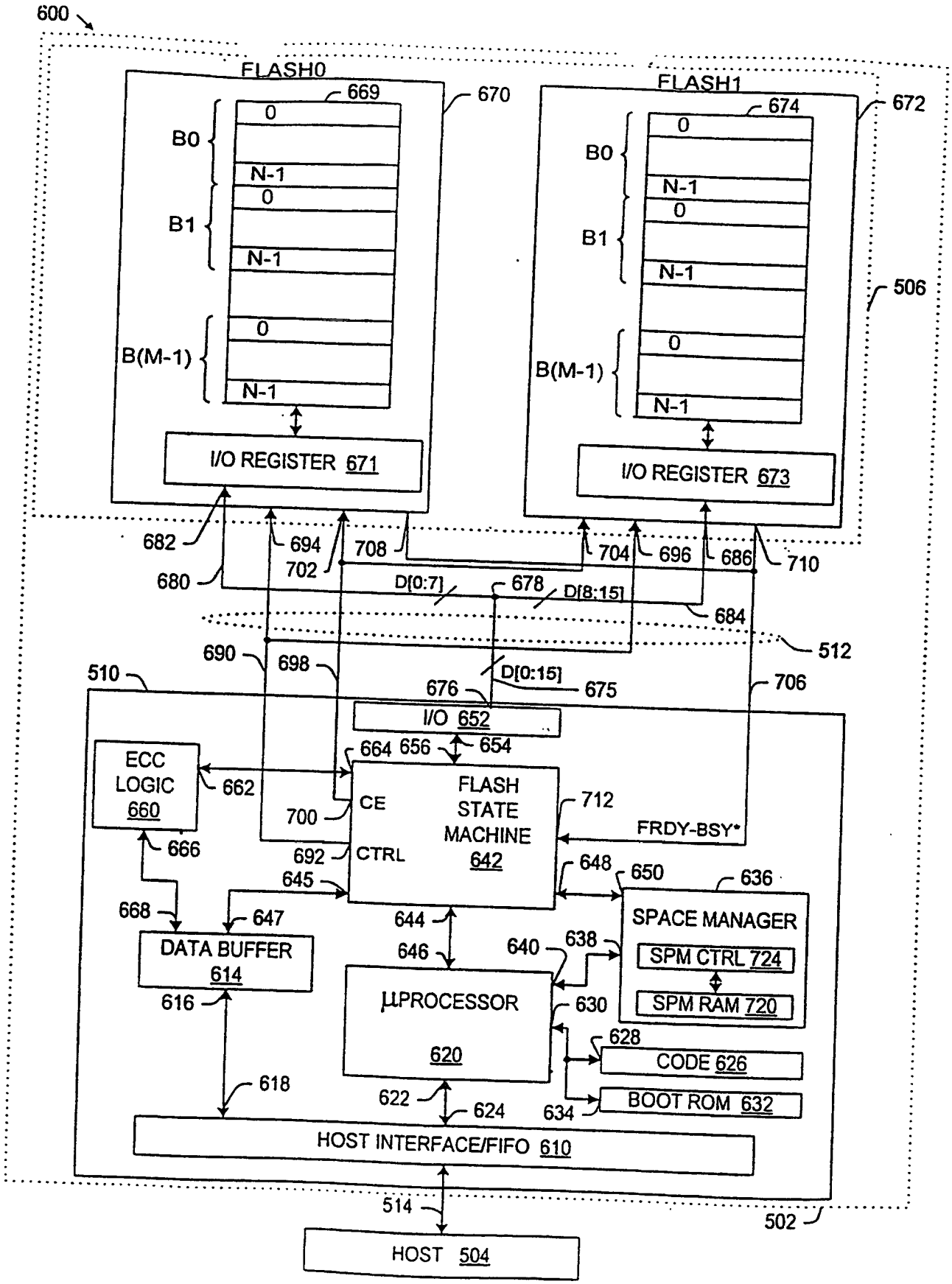

600
FLASH0
669
670
B0
0
N-1
B1
0
N-1
B(M-1)
0
N-1
I/O REGISTER 671
FLASH1
674
672
B0
0
N-1
B1
0
N-1
B(M-1)
0
N-1
I/O REGISTER 673
506
682
694
708
704
696
686
710
680
702
D[0:7]
678
D[8:15]
684
512
690
698
D[0:15]
675
676
706
510
I/O 652
664
656
654
ECC LOGIC 660
662
CE
FLASH STATE MACHINE 642
712
FRDY-BSY*
700
CTRL
692
666
645
648
650
636
668
647
644
SPACE MANAGER
DATA BUFFER 614
646
640
638
SPM CTRL 724
µPROCESSOR 620
630
SPM RAM 720
628
CODE 626
616
618
622
624
634
BOOT ROM 632
HOST INTERFACE/FIFO 610
514
502
HOST 504

FIG. 6

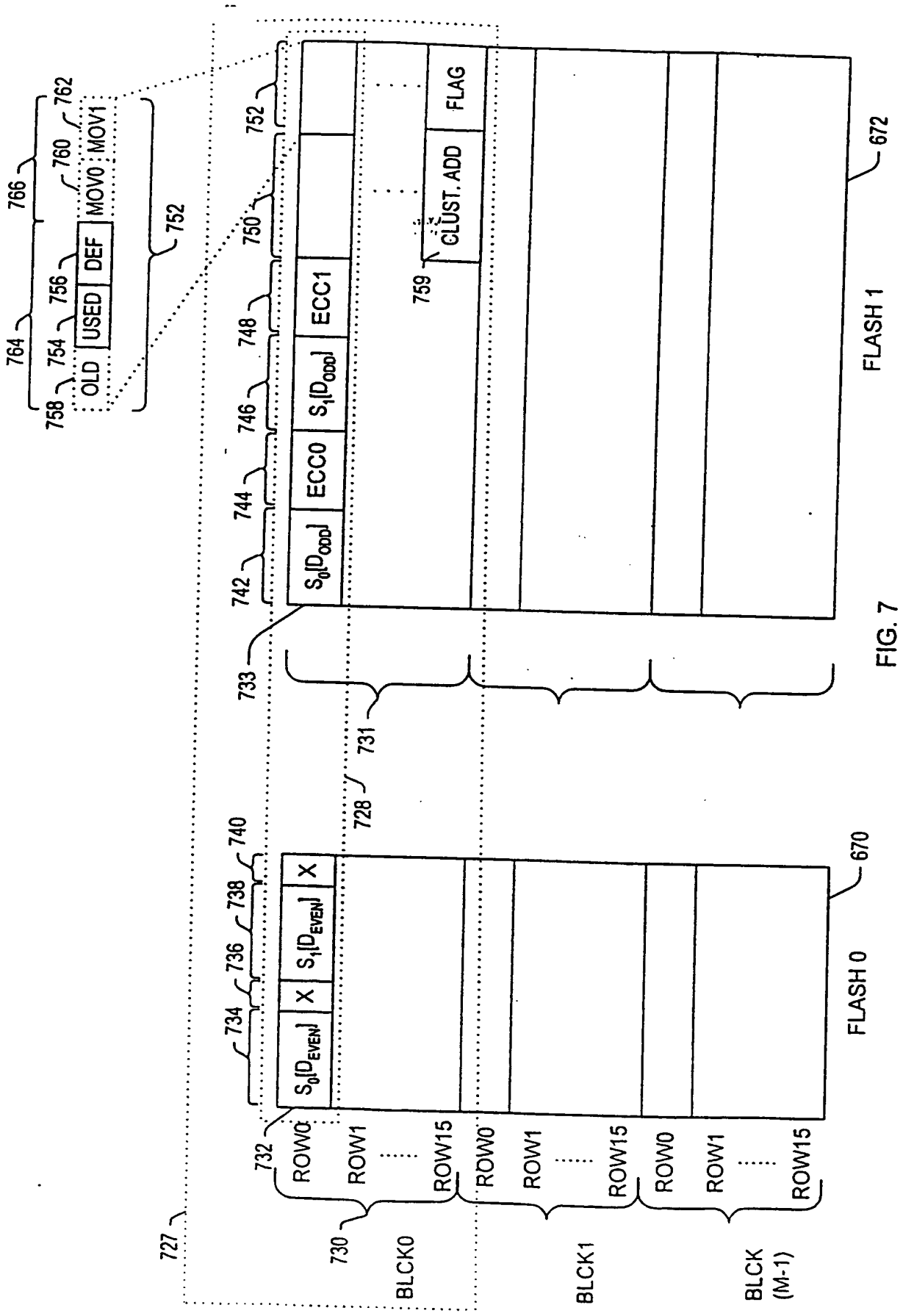


FIG. 7

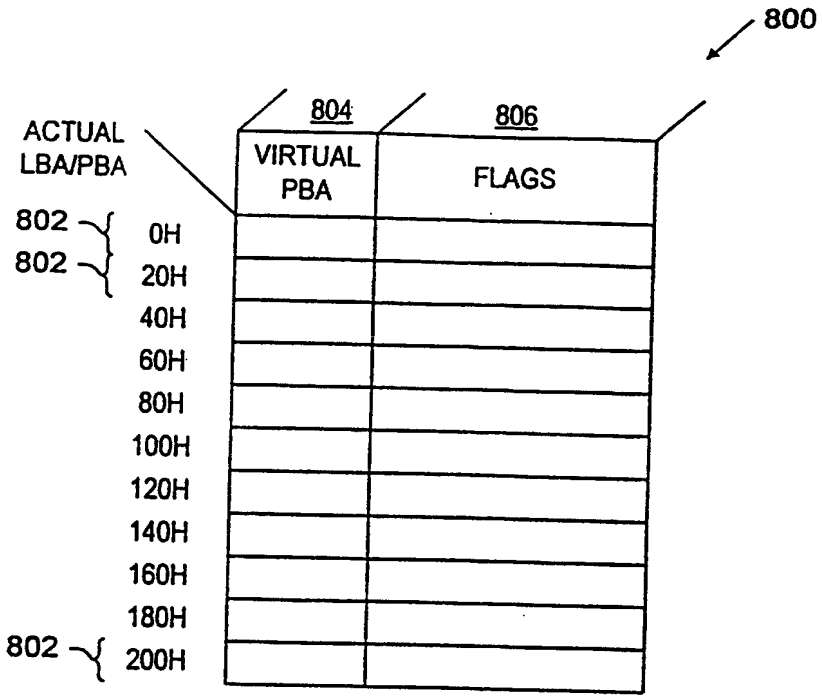


FIG. 8A

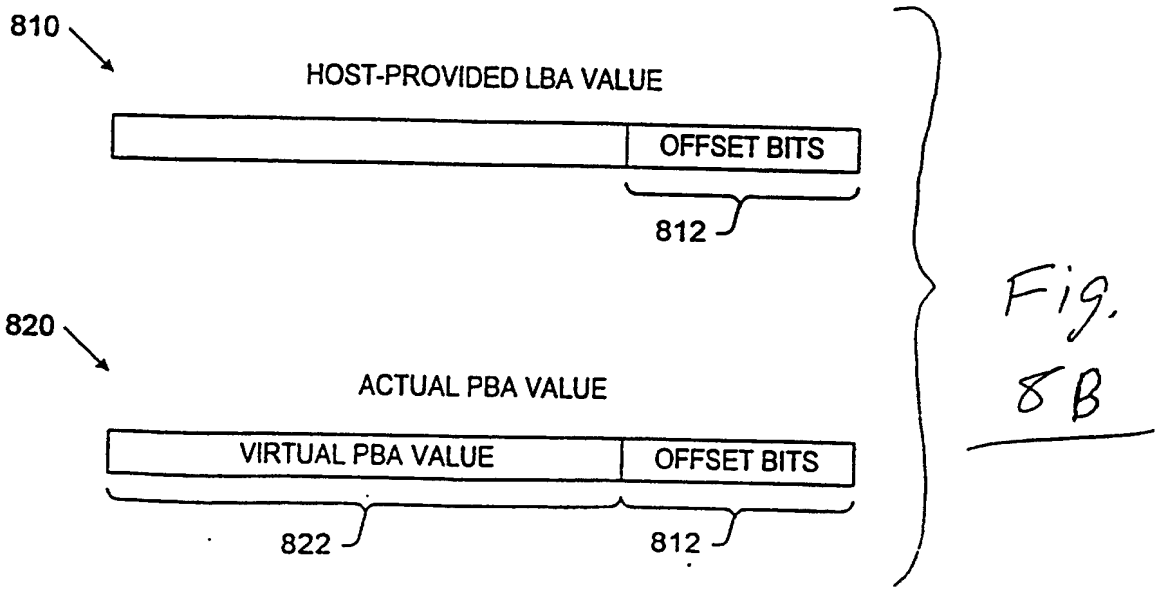


Fig. 8B

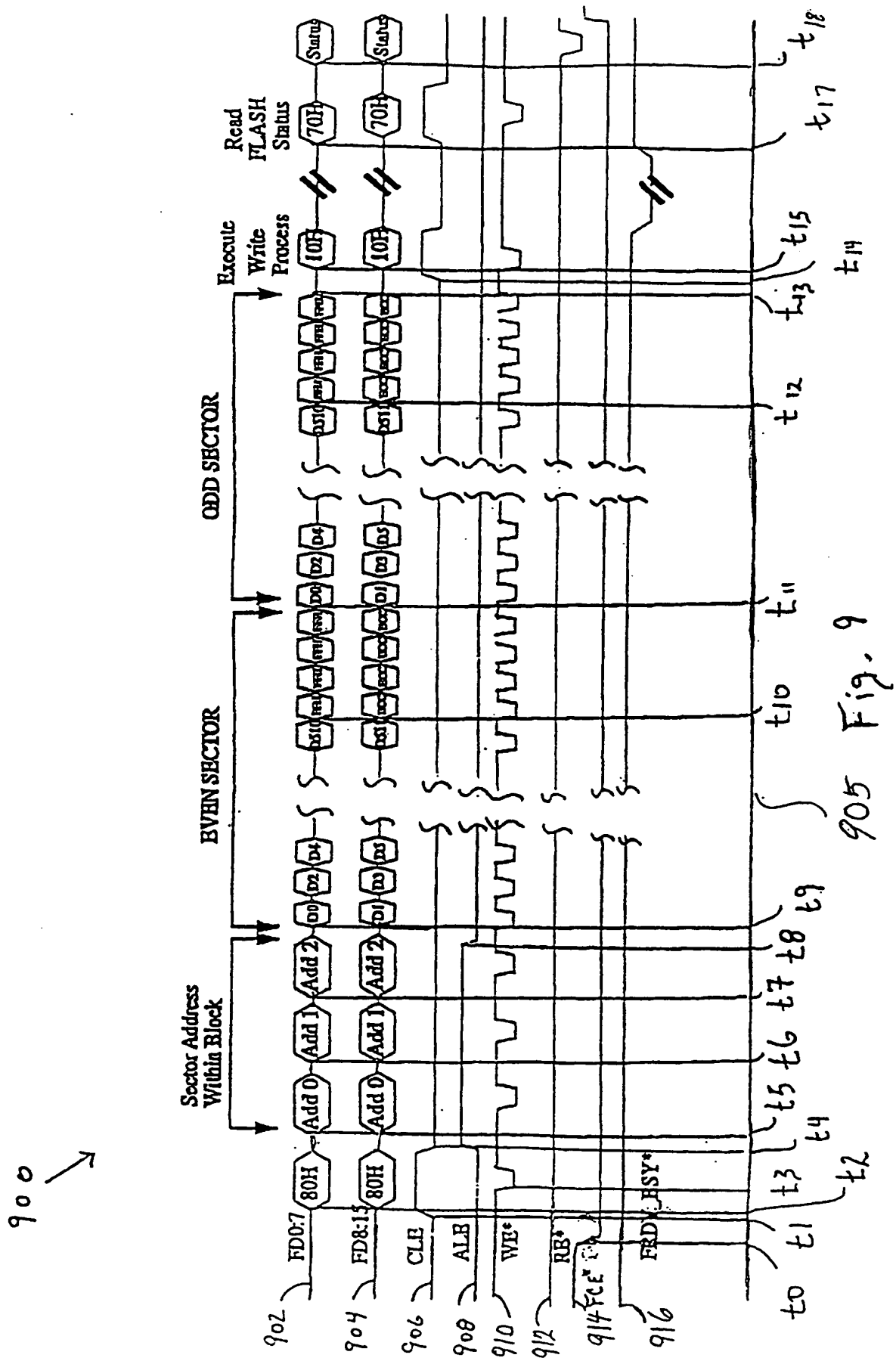


Fig. 9

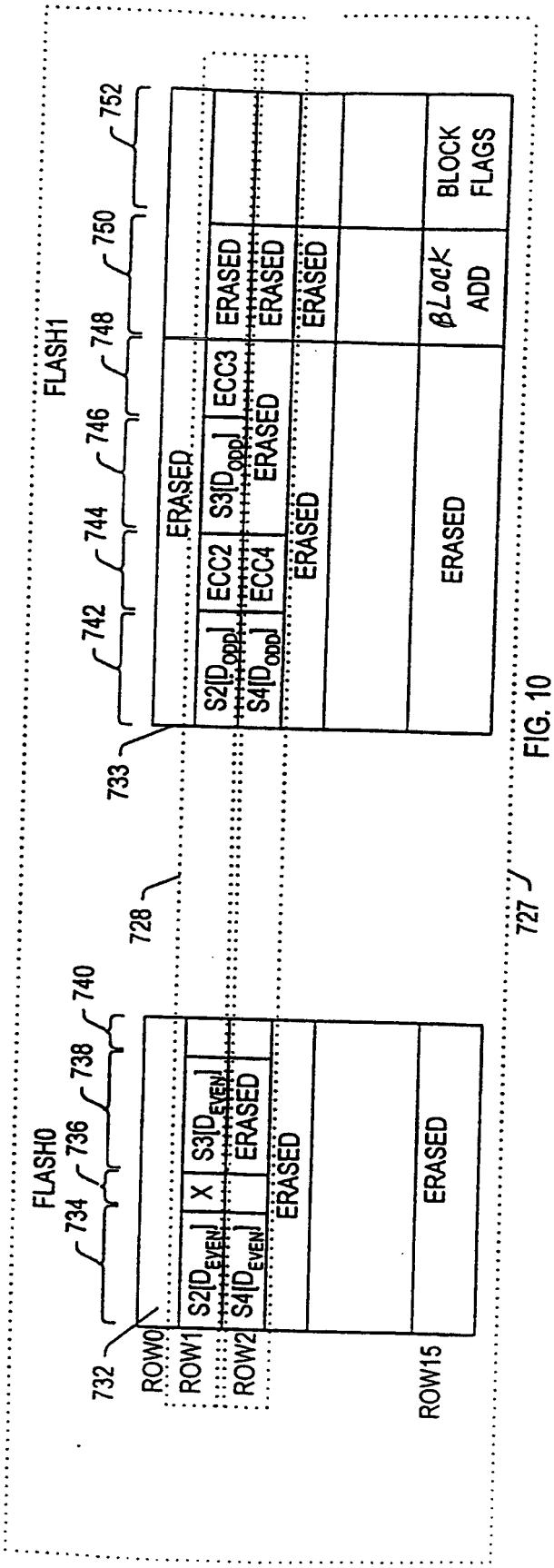


FIG. 10

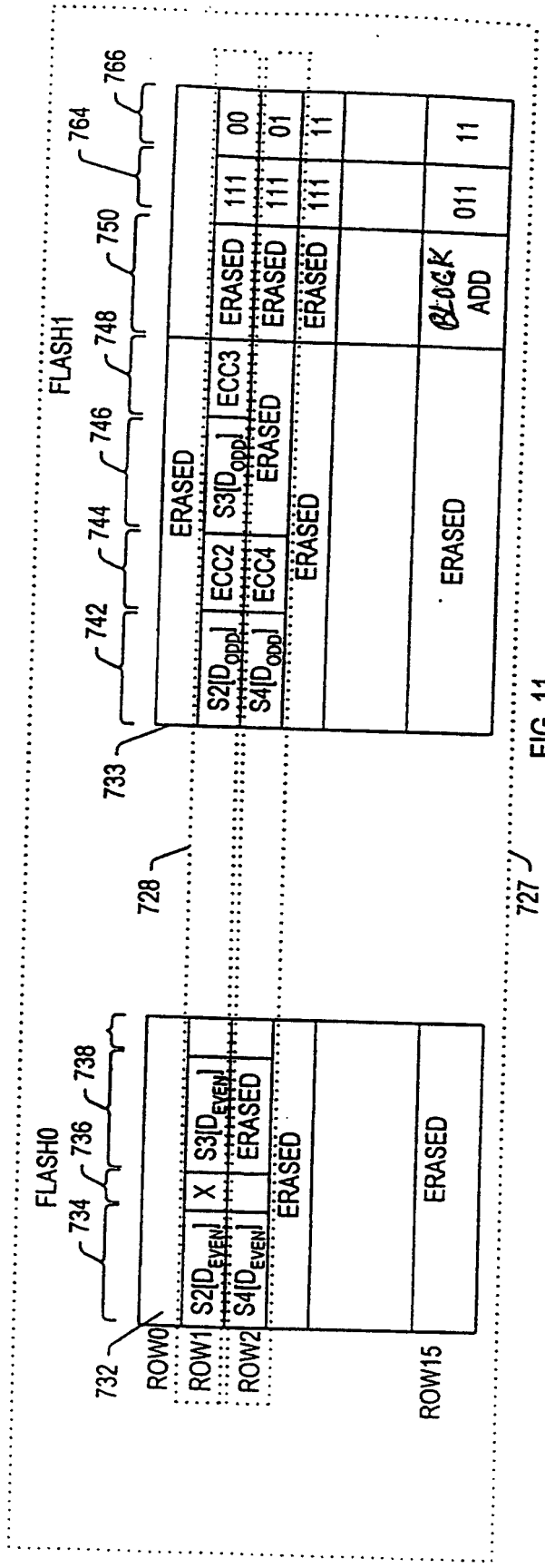


FIG. 11

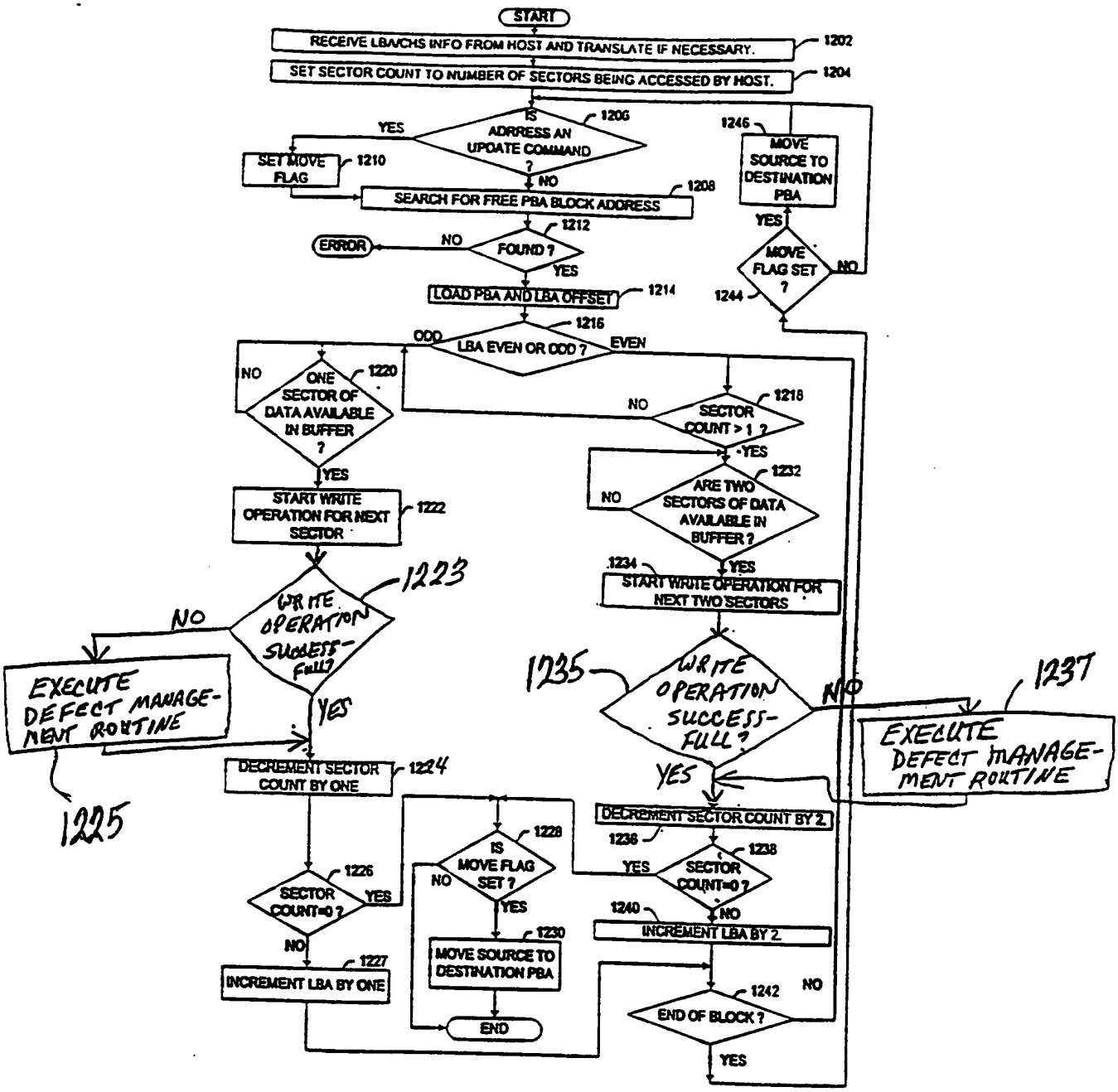

START
RECEIVE LBA/CHS INFO FROM HOST AND TRANSLATE IF NECESSARY.
1202
SET SECTOR COUNT TO NUMBER OF SECTORS BEING ACCESSED BY HOST.
1204
IS ADDRESS AN UPDATE COMMAND ?
1206
YES
NO
SET MOVE FLAG
1210
SEARCH FOR FREE PBA BLOCK ADDRESS
1208
ERROR
NO
FOUND ?
1212
YES
LOAD PBA AND LBA OFFSET
1214
LBA EVEN OR ODD ?
1216
ODD
EVEN
ONE SECTOR OF DATA AVAILABLE IN BUFFER ?
1220
NO
YES
START WRITE OPERATION FOR NEXT SECTOR
1222
WRITE OPERATION SUCCESS-FULL?
1223
NO
YES
EXECUTE DEFECT MANAGE-MENT ROUTINE
1225
DECREMENT SECTOR COUNT BY ONE
1224
SECTOR COUNT=0 ?
1226
YES
NO
INCREMENT LBA BY ONE
1227
IS MOVE FLAG SET ?
1228
NO
YES
MOVE SOURCE TO DESTINATION PBA
1230
END
SECTOR COUNT > 1 ?
1218
NO
YES
ARE TWO SECTORS OF DATA AVAILABLE IN BUFFER ?
1232
NO
YES
START WRITE OPERATION FOR NEXT TWO SECTORS
1234
WRITE OPERATION SUCCESS-FULL ?
1235
NO
YES
EXECUTE DEFECT MANAGE-MENT ROUTINE
1237
DECREMENT SECTOR COUNT BY 2.
1236
SECTOR COUNT=0 ?
1238
YES
NO
INCREMENT LBA BY 2.
1240
END OF BLOCK ?
1242
NO
YES
MOVE SOURCE TO DESTINATION PBA
1246
MOVE FLAG SET ?
1244
YES
NO

Fig. 12

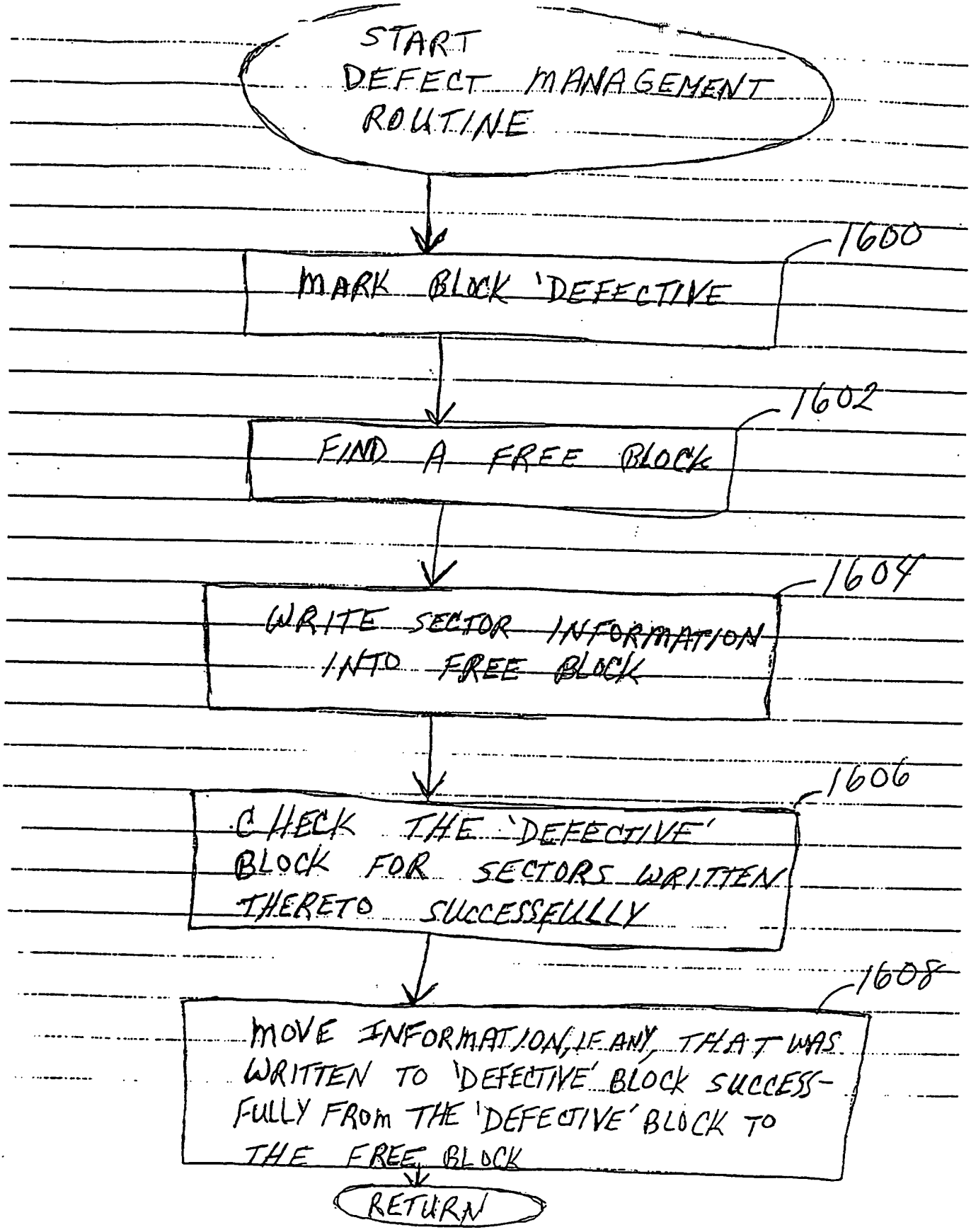


Fig. 12a

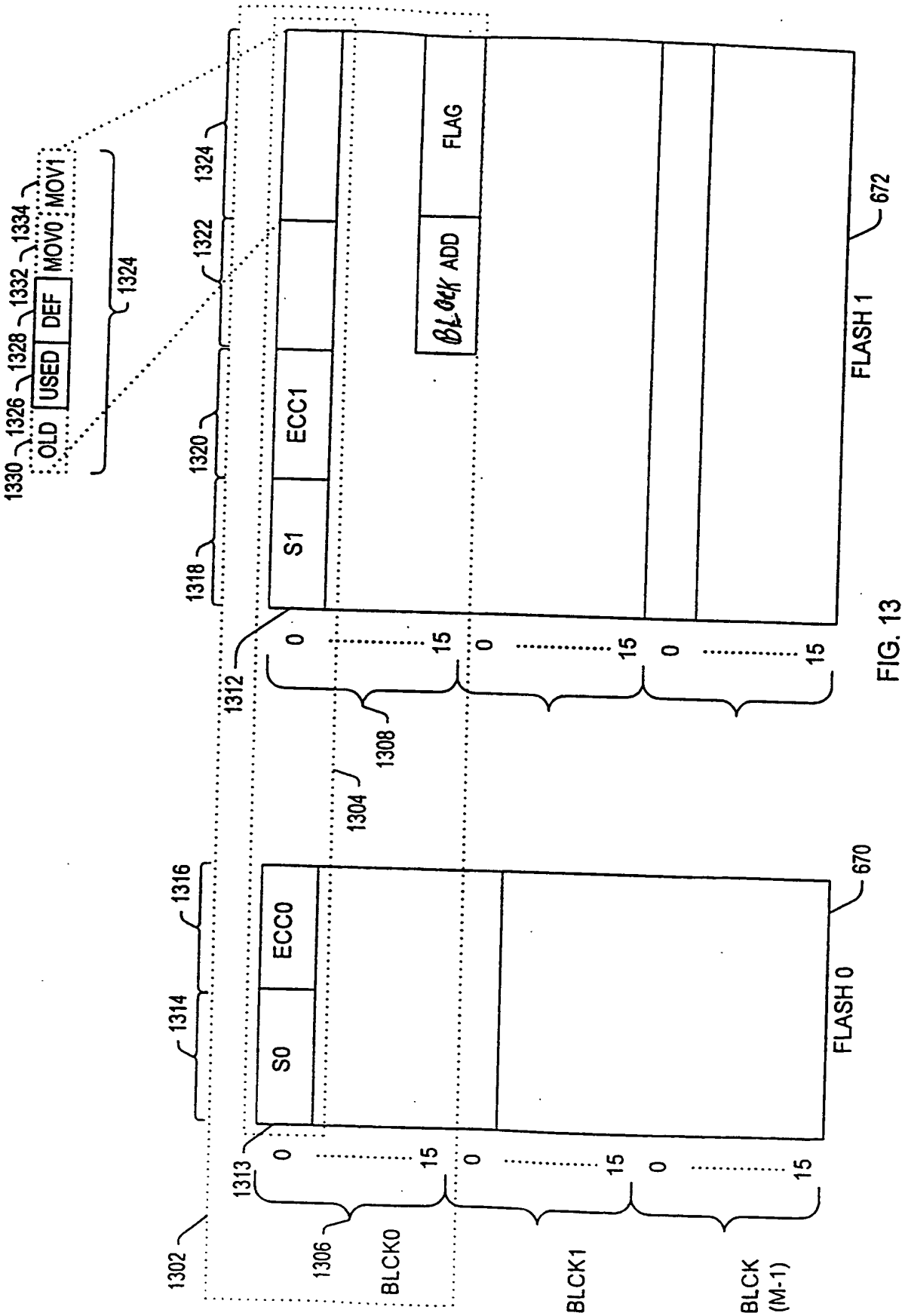

1330
1326
OLD
1328
USED
1332
DEF
1334
MOV0
MOV1
1324
1324
1322
1320
1318
S1
ECC1
FLAG
BLOCK ADD
672
FLASH 1
1312
0
15
0
15
0
15
1308
1304
1316
1314
S0
ECC0
670
FLASH 0
1313
0
15
0
15
0
15
1306
1302
BLCK0
BLCK1
BLCK (M-1)

FIG. 13

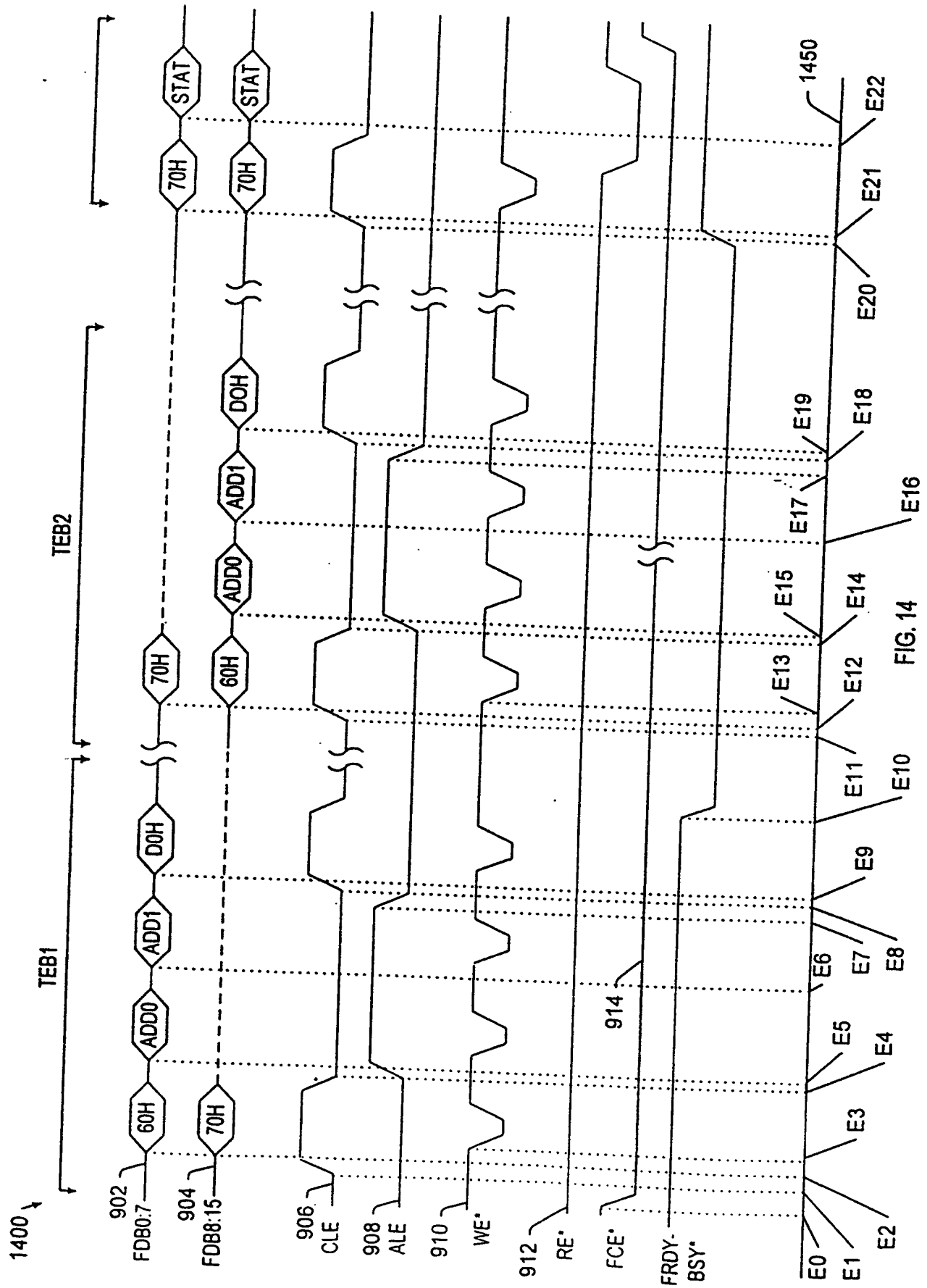


FIG. 14

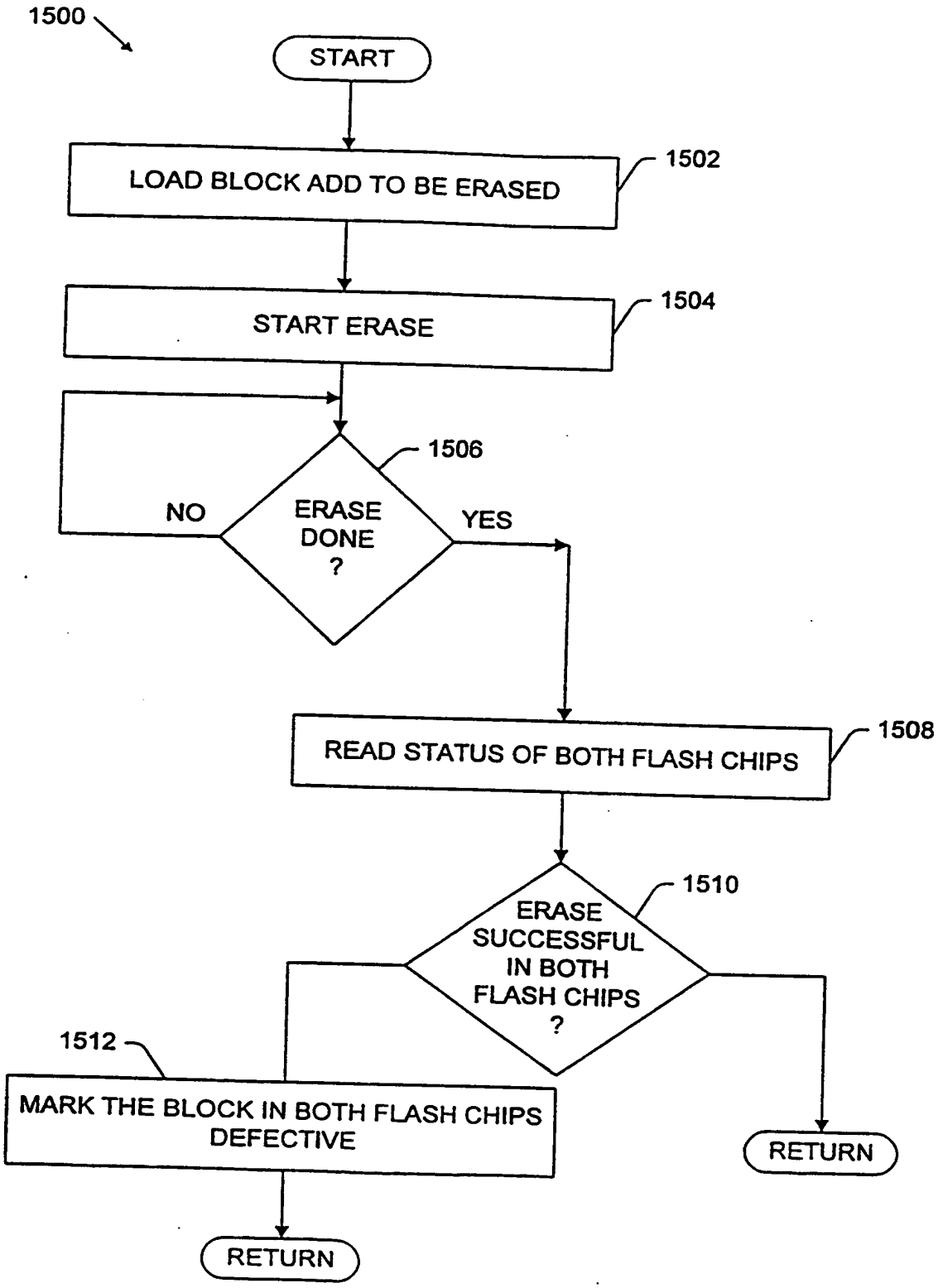


FIG. 15

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 12/00, 12/10 — A3

(11) International Publication Number: WO 99/44113

(43) International Publication Date: 2 September 1999 (02.09.99)

(21) International Application Number: PCT/US99/04247

(22) International Filing Date: 25 February 1999 (25.02.99)

(30) Priority Data:
09/030,697 25 February 1998 (25.02.98) US

(71) Applicant *(for all designated States except US)*: LEXAR MEDIA, INC. [US/US]; 47421 Bayside Parkway, Fremont, CA 94538 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: ESTAKHRI, Petro [US/US]; 7966 Foothills Knolls, Pleasanton, CA 94566 (US). IMAN, Berhanu [US/US]; 946 Iris Avenue, Sunnyvale, CA 94086 (US).

(74) Agents: IMAM, Maryam et al.; Oppenheimer Wolff & Donnelly LLP, Suite 400, 101 Park Center Plaza, San Jose, CA 95113 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

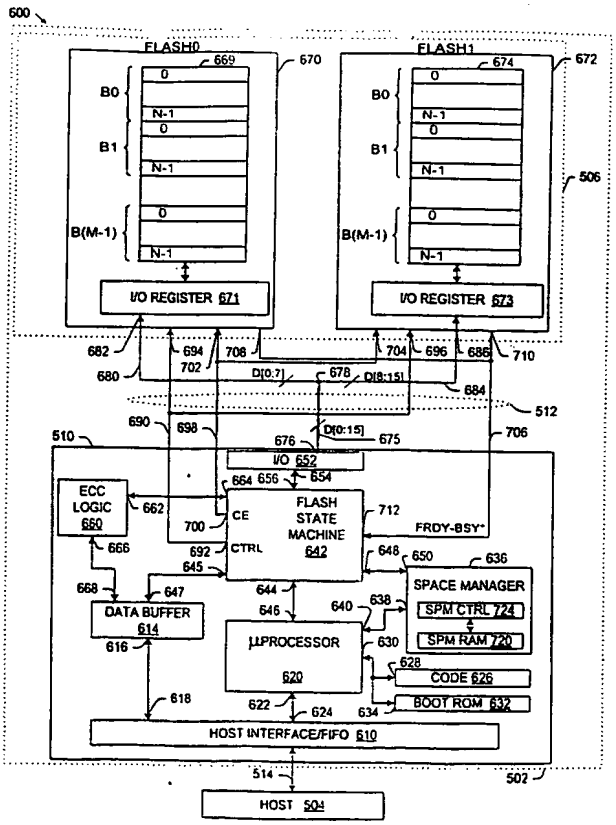
Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(88) Date of publication of the international search report:
7 October 1999 (07.10.99)

(54) Title: INCREASING MEMORY PERFORMANCE IN FLASH MEMORY DEVICES BY PERFORMING SIMULTANEOUS WRITE OPERATION TO MULTIPLE DEVICES

(57) Abstract

The present invention includes a digital system (600) having a controller semiconductor device (510) coupled to a host (504) and a nonvolatile memory bank (506) including a plurality of nonvolatile memory devices (670, 672). The controller transfers information, organized in sectors, with each sector including a user data portion and an overhead portion, between the host and the nonvolatile memory bank and stores and reads two bytes of information relating to the same sector simultaneously within two nonvolatile memory devices. Each nonvolatile memory device is defined by a row of memory locations wherein corresponding rows of at least two semiconductor devices maintain two sectors of information therein with the overhead information related to the two sectors maintained in one of the memory rows of the nonvolatile memory device. Thirty two sectors of information define a block identified by a virtual physical block address with a block of information expanding between two memory devices.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/04247

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :G06F 12/00, 12/10
US CL : 711/103, 168, 173, 202, 206
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 711/103, 168, 173, 202, 206

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, DIALOG
search terms: sectors, concurrent or simultaneous, flash memory

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,375,222 A [ROBINSON et al.] 20 December 1994, C 5, L 39-44; C 25, L 1-19; C 24, L 38-68; C 4, L 60-62; C 11, L 24-68; C 12, L 24-68; C 6, L 56-59; C 7, L 7-35	1-7 12-19, 22-25
A,P	US 5,732,208 A [TAMURA et al] 24 March 1998, C 2, L 65-67; C 3, L 1-32;	1, 12-19, 22-25
X,P	US 5,748,528 A [CAMPARDO et al.] 05 May 1998, C 2, L 12-27	12-14, 22-24
A,E	US 5,890,192 A [LEE et al.] 30 March 1999, C 1, L 27-67	1, 12-19, 22-25

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 11 JUNE 1999	Date of mailing of the international search report 17 AUG 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer KIMBERLY James R. Matthews Telephone No. (703) 308-9592

Form PCT/ISA 210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/04247

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,603,001 A [SUKEGAWA et al.] 11 February 1997, C 4, L 13-30	1, 12-19, 22-25
X,P	US 5,745,418 A [MA et al.], 28 April 1998, C 4, L 24-49; C 8, L 50-67; C 9, L 1-35	9-11, 20-21, 26

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

CORRECTED VERSION*

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 12/00, 12/10

A3

(11) International Publication Number: WO 99/44113

(43) International Publication Date: 2 September 1999 (02.09.99)

(21) International Application Number: PCT/US99/04247

(22) International Filing Date: 25 February 1999 (25.02.99)

(30) Priority Data:
09/030,697 25 February 1998 (25.02.98) US

(71) Applicant *(for all designated States except US)*: LEXAR MEDIA, INC. [US/US]; 47421 Bayside Parkway, Fremont, CA 94538 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: ESTAKHRI, Petro [US/US]; 7966 Foothills Knolls, Pleasanton, CA 94566 (US). IMAN, Berhanu [US/US]; 946 Iris Avenue, Sunnyvale, CA 94086 (US).

(74) Agents: IMAM, Maryam et al.; Oppenheimer Wolff & Donnelly LLP, Suite 400, 101 Park Center Plaza, San Jose, CA 95113 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

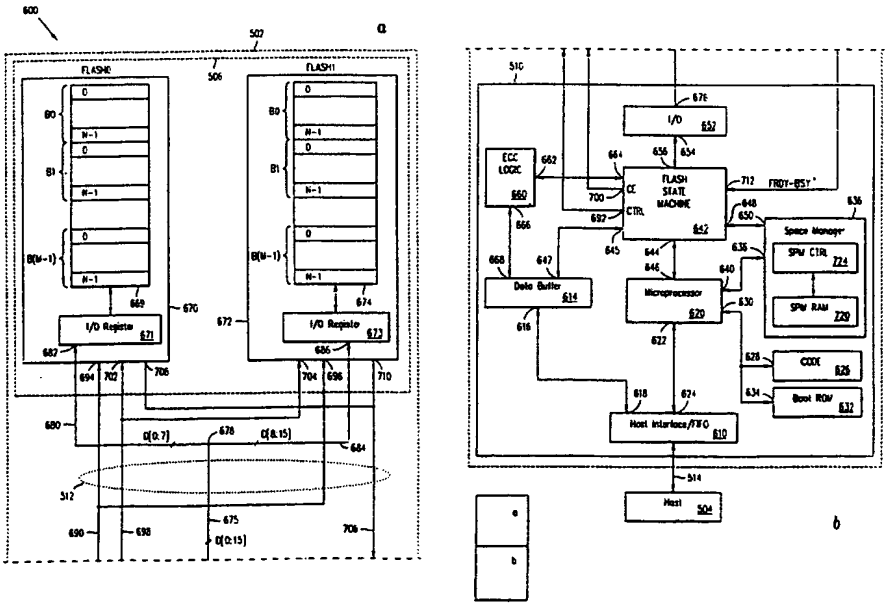
Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(88) Date of publication of the international search report:
7 October 1999 (07.10.99)

(54) Title: INCREASING MEMORY PERFORMANCE IN FLASH MEMORY DEVICES BY PERFORMING SIMULTANEOUS WRITE OPERATION TO MULTIPLE DEVICES

(57) Abstract

The present invention includes a digital system (600) having a controller semiconductor device (510) coupled to a host (504) and a nonvolatile memory bank (506) including a plurality of nonvolatile memory devices (670, 672). The controller transfers information, organized in sectors, with each sector including a user data portion and an overhead portion, between the host and the nonvolatile memory bank and stores and reads two bytes of information relating to the same sector simultaneously within two nonvolatile memory devices. Each nonvolatile memory device is defined by a row of memory locations wherein corresponding rows of at least two semiconductor devices maintain two sectors of information therein with the overhead information related to the two sectors maintained in one of the memory rows of the nonvolatile memory device. Thirty two sectors of information define a block identified by a virtual physical block address with a block of information expanding between two memory devices.



*(Referred to in PCT Gazette No. 45/1999, Section II)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

INCREASING MEMORY PERFORMANCE IN FLASH MEMORY DEVICES BY PERFORMING SIMULTANEOUS WRITE OPERATION TO MULTIPLE DEVICES

BACKGROUND OF THE INVENTION

Cross Reference to Related Applications

This application is a continuation-in-part of our prior application Serial No. 08/946,331 filed October 7, 1997, entitled " Moving Sequential Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", which is a continuation-in-part of application Serial No. 08/831,266, filed March 31, 1997, entitled "Moving Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture".

Field of the Invention

This invention relates to the field of digital systems, such as personal computers and digital cameras, employing nonvolatile memory as mass storage, for use in replacing hard disk storage or conventional film. More particularly, this invention relates to an architecture for increasing the performance of such digital systems by increasing the rate at which digital information is read from and written to the nonvolatile memory.

Description of the Prior Art

With the advent of higher capacity solid state storage devices (nonvolatile memory), such as flash or EEPROM memory, many digital systems have replaced conventional mass storage devices with flash and/or EEPROM memory devices. For example, personal computers (PC s) use solid state storage devices for mass storage purposes in place of

conventional hard disks. Digital cameras employ solid state storage devices in cards to replace conventional films.

Fig. 1 shows a prior art memory system 10 including a controller 12, which is generally a semiconductor (or integrated circuit) device, coupled to a host 14 which may be a PC or a digital camera. The controller 12 is further coupled to a nonvolatile memory bank 16. Host 14 writes and reads information, organized in sectors, to and from memory bank 16 which includes a first nonvolatile memory chip 18 and a second nonvolatile memory chip 20. Chip 18 includes: an I/O register 22 having a port 24 connected to a port 26 of controller 12 via a first bus 28 which includes 8 bit lines; and a storage area 30 coupled with I/O register 22. Chip 20 includes: an I/O register 32 having a port 34 connected to a port 36 of controller 12 via a second bus 38 which includes 8 bit lines; and a storage area 40 coupled with I/O register 32. The first and second buses 28, 38 are used to transmit data, address, and command signals between the controller and the memory chips 18 and 20. The least significant 8 bits (LSBs) of 16 bits of information are provided to chip 18 via the first bus 28, and the most significant 8 bits (MSBs) are provided to the chip 20 via the second bus 38.

Memory bank 16 includes a plurality of block locations 42 each of which includes a plurality of memory row locations. Each block location of the memory bank is comprised of a first sub-block 44 located in the first non-volatile memory chip, and a corresponding second sub-block 46 located in the second non-volatile memory chip. Each memory row location includes a first row-portion 48 and a corresponding second row-portion 50. In the depicted embodiment each of the first and second row-portions 48 and 50 includes storage for 256 bytes of data information plus an additional 8 bytes of storage space for overhead information. Where a sector includes 512 bytes of user data and 16 bytes of non-user data (the latter commonly referred to as overhead information), 256 bytes of the user data and 8 bytes of the overhead information of the sector may be maintained in the first row portion 48 of chip 18 and the remaining 256 bytes of user data and remaining 8 bytes of overhead information of the same sector may be maintained in the second row portion 50 of chip 20. Thus, half of a sector is stored in a memory row location 48 of chip 18 and the other half of the sector is stored in memory row location 50 of chip 20. Additionally, half of the overhead information of each stored sector is maintained by chip 18 and the other half by chip 20.

In general, reading and writing data to flash memory chips 18 and 20 is time consuming. Writing data to the flash memory chips is particularly time consuming because

data must be latched in I/O registers 22 and 32, which are loaded 1 byte at a time via the first and second buses, and then transferred from the I/O registers 22 and 32 to the memory cells of the flash memory chips 18 and 20 respectively. The time required to transfer data from the I/O registers to memory, per byte of data, is proportional to the size of the I/O registers and the size of the flash memory chip.

During a write operation, controller 12 writes a single sector of information to memory bank 16 by: (1) transmitting a write command signal to each of chips 18 and 20 via buses 28 and 38 simultaneously; (2) transmitting address data to chips 18 and 20 specifying corresponding sub-blocks 44 and 46 of the chips via buses 28 and 38 simultaneously; and (3) sequentially transmitting a byte of user data to each of chips 18 and 20 via buses 28 and 38 simultaneously for storage in the corresponding sub-blocks 44 and 46. The problem with such prior art systems is that while two bytes of information are written and read at a time, only one sector of information is accommodated at a time by the memory bank 16 during a write command initiated by the host 14.

Another prior art digital system 60 is shown in Fig. 2 to include a controller 62 coupled to a host 64, and a nonvolatile memory bank 66 for storing and reading information organized in sectors to and from nonvolatile memory chip 68, included in the memory bank 66. While not shown, more chips may be included in the memory bank, although the controller, upon command by the host, stores an entire sector in one chip. A block, such as block 0, includes 16 sectors S0, S1,..., S15. Also included in the chip 68 is an I/O register 70, which includes 512 bytes plus 16 bytes, a total of 528 bytes, of storage space. The controller transfers information between host 64 and memory 66 a byte at-a-time. A sector of 512 bytes of user data plus 16 bytes of overhead information is temporarily stored in the I/O register during a write operation and then transferred to one of the blocks within the memory device for storage thereof. During a read operation, a sector of information is read from one of the blocks of the memory device and then stored in the I/O register for transfer to the controller . An important problem with the prior art architecture of Fig. 2 is that while a total of 528 bytes may be stored in the I/O register 36, only one byte of sector information may be transferred at a time between the controller and the memory bank thereby impeding the overall performance of the system.

Both of the prior art systems of Figs. 1 and 2 maintain LBA to PBA mapping information for translating a host-provided logical block address (LBA) identifying a sector

of information to a physical block address (PBA) identifying the location of a sector within the memory bank. This mapping information may generally be included in volatile memory, such as a RAM, within the controller, although it may be maintained outside of the controller.

Fig. 3 shows a table diagram illustrating an example of an LBA-PBA map 300 defined by rows and columns, with each row 302 being uniquely identified, addressed, by a value equal to that of the LBA received from the host divided by 16. The row numbers of Fig. 3 are shown using hexadecimal notation. Thus, for example, row 10H (in Hex.) has an address value equal to 16 in decimal. Each row 302 of map 300, includes a storage location field 304 for maintaining a virtual PBA value, an 'old' flag field 306, a 'used' flag field 308, and a 'defect' flag field 310. The flag fields provide information relating to the status of a block of information maintained within the memory bank (in Figs. 1 and 2). The virtual PBA field 304 stores information regarding the location of the block within the memory bank.

Fig. 4 shows a table diagram illustrating an exemplary format for storage of a sector of data maintained in a memory bank. The virtual PBA field 304 (Fig. 3) provides information regarding the location of a block 400 of information with each block having a plurality of sectors 402. Each sector 402 is comprised of a user data field 404, an ECC field 406, an 'old' flag field 408, a 'used' flag field 410 and a 'defect' flag field 412.

A further problem associated with prior art systems of the kind discussed herein is that the table 300 (in Fig. 3) occupies much 'real estate' and since it is commonly comprised of RAM technology, which is in itself costly and generally kept within the controller, there is substantial costs associated with its manufacturing. Furthermore, as each row of table 300 is associated with one block of information, the larger the number of blocks of information, the larger the size of the table, which is yet an additional cost for manufacturing the controller and therefore the digital system employing such a table.

What is needed is a digital system employing nonvolatile memory for storage of digital information organized in sector format for reducing the time associated with performing reading and writing operations on the sectors of information thereby increasing the overall performance of the system while reducing the costs of manufacturing the digital system.

SUMMARY OF THE INVENTION

It is an object of the present invention to increase the performance of a digital system having a controller coupled to a host for operating a nonvolatile memory bank including one or more nonvolatile memory devices, such as flash and/or EEPROM chips, by reducing the time associated with reading and writing information to the nonvolatile memory bank.

It is another object of the present invention, as described herein, to decrease the time associated with storing sectors of information by writing at least two sectors of information to at least two nonvolatile memory semiconductor devices during a single write command initiated by the host.

It is another object of the present invention as described herein to decrease the time associated with reading sectors of information by reading at least two sectors of information from at least two nonvolatile memory semiconductor devices during a single read command initiated by the host.

It is a further object of the present invention to store overhead information associated with two sectors of information in one of the two nonvolatile memory semiconductor devices.

It is yet another object of the present invention to simultaneously access two bytes of a sector of information stored within two nonvolatile memory devices thereby increasing the rate of performance of a system employing the present invention by an order of magnitude of at least two.

It is yet another object of the present invention to access one byte of a first sector and one byte of a second sector of information simultaneously within two nonvolatile memory devices thereby increasing the rate of performance of a system employing the present invention.

It is a further object of the present invention to reduce the size of a volatile memory table, or map, that maintains translations between the host-provided sector addresses to addresses of blocks within the nonvolatile memory devices thereby reducing the cost of manufacturing the digital system.

Briefly, the present invention includes a digital system having a controller semiconductor device coupled to a host and a nonvolatile memory bank including a plurality of nonvolatile memory devices. The controller transfers information, organized in sectors,

with each sector including a user data portion and an overhead portion, between the host and the nonvolatile memory bank and stores and reads two bytes of information relating to the same sector simultaneously within two nonvolatile memory devices. Each nonvolatile memory device is defined by a row of memory locations wherein corresponding rows of at least two semiconductor devices maintain two sectors of information therein with the overhead information relating to the two sectors maintained in one of the memory rows of the nonvolatile memory device. Each 32 sectors of information defines a block identified by a virtual physical block address with a block of information expanding between two memory devices wherein an even and an odd byte of a sector is simultaneously read from or written to two nonvolatile memory devices. In another embodiment, the controller stores an entire sector of information within a single nonvolatile memory device and reads from or writes to, a sector of information by processing corresponding bytes of at least two sectors in two nonvolatile memory devices simultaneously.

These and other objects and advantages of the present invention will no doubt become apparent to those skilled in the art after having read the following detailed description of the preferred embodiments illustrated in the several figures of the drawing.

IN THE DRAWINGS

Fig. 1 is a block diagram of a prior art memory system in which a single sector of information is written, two bytes at a time during a write operation, to a memory bank including two memory units each having capacity to store 256 bytes of user data in a single row location;

Fig. 2 is a block diagram of a prior art memory system in which a single sector of information is written, one byte at a time during a write operation, to a memory bank including at least one memory unit having capacity to store 512 bytes of user data in a single row location;

Fig. 3 is a table diagram illustrating an exemplary map for translating a host-provided logical block address (LBA) identifying a sector of information to a physical block address (PBA) identifying a location for the sector within a memory bank;

Fig. 4 is a table diagram illustrating an exemplary format for storage of a sector of data maintained in a memory bank;

Fig. 5 is a generalized block diagram of a memory system in accordance with the present invention in which two sectors of information are written, two bytes at a time during a single write operation, to a memory bank including at least two memory units each having capacity to store 512 bytes of user data in a single row location;

Figure 6 is a detailed block diagram of the memory system of Fig. 5;

Fig. 7 is a table diagram generally illustrating a memory storage format for storing a block, including 32 sectors, of information in a memory bank including two non-volatile memory units wherein an even sector and an odd sector are stored in a single memory row location and wherein even data bytes of both sectors are stored in a row portion located in a first of the memory units and odd data bytes of both sectors are stored in a second row portion located in the second of the memory units;

Fig. 8A is a table diagram generally illustrating organization of an exemplary LBA-PBA map for use in accordance with the present invention;

Fig. 8B shows a block diagram illustrating formats of address information identifying sectors and associated blocks of information in accordance with the present invention;

Fig. 9 is a timing diagram illustrating the timing of control, address, and data signals for a write operation performed by the memory system of Fig. 6 wherein two sectors of information are simultaneously written, during a single write operation, to a memory bank having the memory storage format illustrated in Fig. 7;

Fig. 10 is a table diagram illustrating a memory bank having a memory storage format as depicted in Fig. 7 wherein a single sector is written to a particular memory row location of the memory bank;

Fig. 11 is a table diagram illustrating a memory bank having an alternative memory storage format as depicted in Fig. 7 wherein a single sector is written to a particular memory row location of the memory bank;

Fig. 12 is a flowchart illustrating a process of simultaneously writing two sectors of information to two memory units during a single write operation in accordance with the present invention;

Fig. 12a shows a flow chart of the steps performed in executing the defect management routine of Fig. 12.

Fig. 13 is a table diagram generally illustrating an alternative memory storage format for storing a block, including 32 sectors, of information in a memory bank including two non-volatile memory units wherein an even sector and an odd sector are stored in a single memory row location and wherein an even sector is stored in a first row portion located in a first of the two memory units and an odd sector is stored in a second row portion located in the second of the two memory units;

Fig. 14 shows a timing diagram illustrating the timing of control, address, and data signals for a process of erasing a block of a memory bank in accordance with principles of the present invention; and

Fig. 15 is a flowchart illustrating a process of erasing a block, including a first sub-block stored in a first memory unit and a second sub-block stored in a second memory unit, in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 5 shows a generalized block diagram at 500 of a memory system in accordance with principles of the present invention. The system includes a memory card 502 coupled to a host system 504. In one embodiment, host 504 is a digital camera and memory card 502 is a digital film card, and in another embodiment, host 504 is a personal computer system and memory card 502 is a PCMCIA card. Memory card 502 includes: a non-volatile memory bank 506 including a plurality of non-volatile memory units 508 for storing sectors of information organized in blocks; a memory controller 510 coupled to the memory bank via a memory bus 512, and coupled to the host 504 via a host bus 514. Memory controller 510 controls transfer of sector-organized information between host 504 and memory bank 506. Each sector of information includes a user data portion and an overhead portion. The memory controller performs write and read operations, in accordance with the present invention, to and from the memory units of the memory bank as further explained below.

In the present invention, the non-volatile memory bank 506 may include any number of non-volatile memory units 508 while in a preferred embodiment, the non-volatile memory bank has an even number of memory units. Also in the preferred embodiment, each of the non-volatile memory units is a flash memory integrated circuit device.

Fig. 6 shows a detailed block diagram at 600 of the memory system of Fig. 5. Controller 510 is shown to include: a host interface 610 connected to the host 504 via host bus 514 for transmitting address, data, and control signals between the controller and the host; a data buffer 614 having a port 616 coupled to a port 618 of the host interface; a microprocessor 620 having a port 622 coupled to a port 624 of the host interface; a code storage unit 626 having a port 628 coupled to a port 630 of the microprocessor; a boot ROM unit 632 having a port 634 coupled to port 630 of the microprocessor and to port 628 of the code storage unit; a space manager 636 having a port 638 coupled to a port 640 of the microprocessor; a flash state machine 642 including a port 644 coupled to a port 646 of the microprocessor, a port 648 coupled to a port 650 of the space manager, and a port 645 coupled to a port 647 of the data buffer; a memory input/output unit 652 having a port 654 coupled to a port 656 of the flash state machine; an error correction code logic unit (ECC logic unit) 660 having a port 662 coupled to a port 664 of the flash state machine, and a port 666 coupled to a port 668 of the data buffer 614.

In the depicted embodiment, memory bank 506 includes two non-volatile memory units (although additional memory units may be included, only two are shown for simplicity); a first flash memory chip 670 designated FLASH0 and a second flash memory chip 672 designated FLASH1. First flash memory chip 670 includes a first input/output register (first I/O register) 671 and a storage area 669. Second flash memory chip 672 includes a second input/output register (second I/O register) 673 and a storage area 674.

Memory bus 512 is used to transmit address, data, and control signals between the controller 510 and memory bank 506. Memory bus 512 includes a flash bus 675 connected to a port 676 of memory I/O unit 652 for transmitting address, data, and command signals between flash memory chips 670, 672 and the memory I/O unit 652. Flash bus 675 includes 16 bit lines, 8 bit lines of which form a first bus 680 connected to a port 682 of I/O register 671 of the first flash memory chip, and another 8 bit lines of which form a second bus 684 connected to a port 686 of I/O register 673 of the second flash memory chip.

Memory bus 512 also includes: a control bus 690 which connects a control signal (CTRL signal) output 692 of the flash state machine 642 to an input 694 of the first flash memory chip and to an input 696 of the second flash memory chip; a chip enable line 698 which connects a chip enable (CE) output 700 of the flash state machine 642 to an enable input 702 of the first flash memory chip and to enable an input 704 of the second flash

memory chip; and a ready/busy signal (FRDY-BSY* signal) line 706 which connects an output 708 of the first flash memory chip and an output 710 of the second flash memory chip to an input 712 of the flash state machine 642.

Microprocessor 620, at times (for example, during initialization of the memory system), executes program instructions (or code) stored in ROM 632, and at other times, such as during operation of the memory system, the microprocessor executes code that is stored in code storage unit 626, which may be either a volatile, i.e., read-and-write memory (RAM) or a non-volatile, i.e., EEPROM, type of memory storage. Prior to the execution of program code from code storage unit 626, the program code may be stored in the memory bank 506 and later downloaded to the code storage unit for execution thereof. During initialization, the microprocessor 620 can execute instructions from ROM 632.

Sector-organized information, including user data and overhead information, is received at host interface 610 from host 504 via host bus 514 and provided to the data buffer 614 for temporary storage therein. Sectors of information stored in the data buffer are retrieved under control of flash state machine 642 and provided to memory bank 506 in a manner further described below. It is common in the industry for each sector to include 512 bytes of user data plus overhead information. Although a sector may include other numbers of bytes of information, in the preferred embodiment, a sector has 512 bytes of user data and 16 bytes of overhead information.

ECC logic block 660 includes circuitry for performing error coding and correction on the sector-organized information. ECC logic block 660 performs error detection and/or correction operations on the user data portions of each sector stored in the flash memory chips 670, 672 or data received from host 504.

When required, the space manager 636 finds a next unused (or free) non-volatile memory location within the memory bank for storing a block of information with each block including multiple sectors of information. In the preferred embodiment, a block includes 32 sectors although, alternatively a block may be defined to include another number of sectors such as, for example, 16. The physical address of a storage block located within memory bank 506, referred to as a virtual physical block address (virtual PBA), and the physical block address of a sector of information located within the memory bank 506, referred to as an actual physical block address (actual PBA), is determined by the space manager by performing a translation of a logical block address (LBA) received from the host. An actual

LBA received from host 504 (a host-provided LBA) identifies a sector of information. Space manager 636 includes a space manager memory unit, which is preferably a volatile memory unit, for storing an LBA-PBA map for translating a modified version of the host-provided LBAs to virtual PBAs as further explained below. In the depicted embodiment, the space manager includes a space manager RAM unit (SPM RAM unit) 720 for storing the LBA-PBA map under the control of a space manager controller (SPM controller) 724 which is coupled to the SPM RAM unit.

Fig. 7 shows a table diagram generally illustrating organization of user data, error correction information, and flag information stored in memory bank 506 in accordance with an embodiment of the present invention. Memory bank 506 includes a plurality of M blocks 727 designated BLCK0, BLCK1, BLCK(M-1), each having a virtual physical block addresses (PBA). Each of the blocks 727 includes a plurality of N memory row locations 728 designated ROW0, ROW1,...ROW15 where, in the preferred embodiment, N=16. Each block 727 of memory bank 506 is comprised of a first sub-block 730 of first flash memory chip 670, and a corresponding second sub-block 731 of second flash memory chip 672. Corresponding sub-blocks 730, 731, which together form a block, are identified by the same virtual PBA. Each memory row location 728 includes a first row-portion 732 and a corresponding second row-portion 733 In the depicted embodiment each of the first and second row-portions 732, 733 includes storage for 512 bytes of data information plus additional storage space for other information. In the depicted embodiment, the storage of information in the first row-portions 732 of the first flash memory chip is accomplished in a manner dissimilar from that in the second row-portions 733 of the second flash memory chip.

Each of the first row-portions 732 includes: a first even sector field 734 for storing even data bytes D0, D2, D4,...D510 of an even sector (S0, S2, S4,...) of information; a first spare field 736; a first odd sector field 738 for storing even data bytes D0, D2, D4,...D510 of an odd sector (S1, S3, S5,...) of data; and a second spare field 740. Each of the second row-portions 733 includes: a second even sector field 742 for storing odd data bytes D1, D3, D5,...D511 of the even sector of data which has it's corresponding even data bytes stored in first even sector field 734; a first error correction field 744 for storing error correction information corresponding to the even sector of information stored collectively in fields 734 and 742; a second odd sector field 746 for storing odd data bytes of the odd sector of information which has it's even data bytes stored in first odd sector field 738; a second error

correction field 748 for storing ECC information corresponding to the odd sector of information stored collectively in fields 738 and 746; a block address field 750; and a flag field 752. Fields 734 and 742 form an even sector location while fields 738 and 746 form an odd sector location. It is understood in the present invention that fields 734 and 742 could alternatively form an odd sector location while fields 738 and 746 could alternatively form an even sector location, and that fields 734 and 738 could alternatively be used to store odd data bytes while fields 742 and 746 could alternatively be used to store even data bytes. Additionally, first row-portion 732 could alternatively be used for storing the overhead information relating to the sectors stored in the memory row location 728.

Flag field 752 is used for storing flag information which is used by controller 510 (Fig. 6) during access operations as further explained below. Block address field 750 is used for storing a modified version of a host-provided LBA value which is assigned to a block, as further described below. Only a single block address entry is required in the block address field per block. In a preferred embodiment, a modified host-provided LBA value is entered in block address field 759 of the Nth row, ROW15, of the row locations 728 of each block 727.

In operation, the controller 510 (Fig. 6) accesses an even sector of information stored collectively in the first and second flash memory chips by simultaneously accessing first and second even sector fields 734, 742 of corresponding row-portions of the first and second flash memory chips via the first and second split buses 680, 684 (Fig. 6), respectively. The first and second split buses 680, 684 (Fig. 6) include lines coupled to receive the even and odd data bytes respectively of a sector of information. The controller 510 (Fig. 6) accesses an odd sector of information stored collectively in the first and second flash memory chips by simultaneously accessing the first and second odd sector fields 738, 746 via the first and second split buses 680, 684 (Fig. 6), respectively. The split buses 680, 684 (Fig. 6) also provide for: transmission of ECC information between the flash memory chips and the flash state machine 642 and ECC logic unit 660 of the memory controller 510; and transmission of address information from flash state machine 642 to the flash memory chips.

Controller 510 (Fig. 6) monitors the status of blocks 727 of memory bank 506 using the space manager 636. In one embodiment, controller 510 (Fig. 6) monitors the status of each block location 727 of the memory bank using block level flags including a used/free block flag and a defect block flag stored in a used flag location 754 and a defect flag location 756 respectively of the flag field 752. Block level flags provide information concerning the

status of a whole block 727 of the memory bank and therefore, only a single block level flag entry is required in the flag locations 754 and 756 per block. The used/new block flag indicates whether the corresponding block 727 is currently being "used" to store information or whether it is available (or free) to store information. The defect block flag indicates whether the corresponding block 727 is defective.

In another embodiment, controller 510 (Fig. 6) monitors the status of each memory row location 728 of the memory bank using flags including a used/free row flag stored in the used flag location 754, a defect row flag stored in the defect flag location 756, an old row flag stored in an old flag location 758 of the flag field 752, an even sector move flag stored in an even sector move flag location 760, and an odd sector move flag stored in an odd sector move flag location 762. In this embodiment, the used/new flag indicates whether the corresponding memory row location 728 is currently being "used" to store information or whether it is available (or free) to store information. The defect flag indicates whether the memory block 727 is defective. If either of a corresponding pair of non-volatile memory locations 732, 733 is determined to be defective, then the whole memory block 727 is declared to be defective as indicated by the value in the defect flag location 756 being set, and the defective block can no longer be used. In a preferred embodiment, locations 758, 754, and 756 are included in a single 3-bit flag location 764.

The even and odd sector move flag locations 760, 762 store values indicating whether the corresponding even and odd sectors stored in the non-volatile memory sector location have been moved to another location within the non-volatile memory bank 506 (Fig. 6). For example, if an even sector of information stored collectively in a particular pair of even sector fields 734 , 742 of a row location 728 has been moved to another pair of even sector fields in the non-volatile memory bank 506, the value in the corresponding even sector move flag location 760 is set. Similarly, if an odd sector of information stored collectively in the odd sector fields 738, 746 of the same row location has been moved to another pair of odd sector fields in the non-volatile memory bank, then the value in the corresponding odd sector move flag location 672 is set. The location within the non-volatile memory bank 506 to which a sector of information has been moved is indicated in the LBA-PBA map stored in the SPM RAM 720 in an MVPBA address location, as taught in a patent application, filed by the inventors of this application, entitled "Moving Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", Serial No. 08/831,266, filed March 31, 1997, the

disclosure of which is incorporated herein by reference. In a preferred embodiment, locations 760 and 762 are formed by a single 2-bit move-flag location 766.

Fig. 8A shows a table diagram generally illustrating organization of an exemplary LBA-PBA map at 800, which is stored in SPM RAM 720 (Fig. 6), for translating a modified version of the host-provided LBA's to PBA's. The modified host-provided LBA is derived by dividing the host-provided LBA by the number of sectors with a block, as explained in more detail below. The depicted LBA-PBA map includes: a plurality of map row locations 802 which are addressable by a modified host-provided LBA or by a virtual PBA; a virtual PBA field 804 for storing a virtual PBA value identifying a block 727 (Fig. 7) within the memory bank; and a flag field 806 for storing flag information. As previously mentioned, the actual PBA specifies the location of a sector of information in the memory bank and the virtual PBA specifies the location of a block 727 (Fig. 7) in the memory bank. Virtual PBA values are retrieved by space manager 636 (Fig. 7) from the depicted map and transferred to port 648 of the flash state machine 642 for use in addressing blocks within memory bank 506.

Fig. 8B shows a block diagram illustrating a host-provided-LBA format 810 and an actual PBA format 820. LBA format 810 includes "offset bits" 812, which comprise the least significant bits of the host-provided LBA value. As explained above, in the preferred embodiment, each block 727 (Fig. 7) includes memory space for storing 32 sectors of information, each sector includes 512 bytes of user data and 16 bytes of overhead information. Because each block 727 (Fig. 7) includes 32 sectors in the preferred embodiment, five offset bits 812 are required to identify each of the 32 sectors in each block. In this embodiment, the translation of the host-provided-LBA to actual and virtual PBA values is performed by first masking the five least significant "offset" bits 812, of the host-provided-LBA, shifting the result to the right by 5 bits and using the shifted value as a modified host-provided LBA value or an "LBA-map-value" to address a map row location 802 in the LBA-PBA map 800 (Fig. 8A). This, in effect, is dividing the host-provided LBA by 32. The actual PBA value 820, which specifies the location of a sector within a block of the memory bank, is formed by concatenating offset bits 812 of the LBA value with a virtual PBA 822 value stored in the corresponding field 804 (Fig. 8A) of the LBA-PBA map. That is, the virtual PBA value 822 is used to identify a block within the memory bank and the five remaining offset bits 812 are used to address a sector within the identified block.

Upon initialization of memory system 600 (Fig. 6), the virtual PBA value stored in the virtual PBA field 804 of each map row location 802 is set to an all '1's state. Each time a block 727 (Fig. 7) is accessed by the controller, such as during a write operation, the virtual PBA value stored in the corresponding virtual PBA field 804 of the corresponding map row location is modified by the space manager controller 724 (Fig. 6) to specify a new virtual PBA value. When a block within the memory bank 506 is erased, the old virtual PBA value (the virtual PBA value corresponding to the erased block), rather than a modified version of the host-provided LBA, is used to address the SPM RAM 720 (Fig. 6) and the used flag, stored within the flag field of the SPM RAM 720, is cleared. This same 'used' flag within the flag field of the SPM RAM 720 is set at the time when the corresponding virtual PBA is updated pointing to the new block in the memory bank where sector information is maintained (step 1214).

Fig. 9 shows a timing diagram illustrating the timing of control, address, and data signals for a write operation performed by memory system 600 (Fig. 6) wherein two sectors of information are simultaneously written in the non-volatile memory bank 506 (Fig. 6) during a single write operation. The diagram includes: a wave form 902 representing a first flash signal which transmits time multiplexed command, address, and data information from flash state machine 642 (Fig. 6) of the controller via bus 680 (Fig. 6) to port 682 of the first flash memory chip; a wave form 904 representing a second flash signal which transmits time multiplexed command, address, and data signals from the flash state machine via bus 684 (Fig. 6) to port 686 of the second flash memory chip; a time line 905; and a plurality of control signal wave forms.

The control signal wave forms include: a wave form 906 representing a command line enable signal (CLE signal) transmitted from flash state machine 642 (Fig. 6) to the first and second flash memory chips via control bus 690 (Fig. 6); a wave form 908 representing an address line enable signal (ALE signal) transmitted from the flash state machine to the flash memory chips via the control bus; a wave form 910 representing a write enable signal (WE signal) transmitted from the flash state machine to the flash memory chips via the control bus; a wave form 912 representing a read enable signal (RE signal) transmitted from the flash state machine to the memory chips via the control bus; a wave form 914 representing a flash chip enable signal (FCE* signal) transmitted from chip enable signal output 700 (Fig. 6) of the flash state machine via chip enable line 698 to the first and second flash memory chips; a

wave form 916 representing a flash ready/busy signal (FRDY_BSY* signal) transmitted from outputs 708 and 710 (Fig. 6) of the first and second flash memory chips to the flash state machine via flash ready/busy signal line 706.

The write operation commences at a time t0 at which the FCE* signal (wave form 914) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to begin receiving command, address, data, and control signals. Prior to time t0, the FRDY_BSY* signal (wave form 916), transmitted from the flash memory chips to input 712 of the flash state machine (Fig. 6), is already activated indicating that the first and second flash memory chips are ready to receive access commands. At a subsequent time t1, the CLE signal (wave form 906) is activated, transitioning from a LOW state to a HIGH state, thereby enabling the first and second flash memory chips to read command signals. At a time t2, the first and second flash signals (wave forms 902 and 904) simultaneously transmit a serial data shift-in command signal 80H to the first and second flash memory chips via the first and second first split buses 680 and 684 respectively. At a time t3, while the serial data shift-in command signals 80H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the serial data command signals 80H. At a time t4, the CLE signal (wave form 906) is deactivated, transitioning back to the LOW state, thereby disabling the flash memory chips from reading command signals.

Also at time t4, the ALE signal (wave form 908) is activated, transitioning from a LOW state to a HIGH state, thereby enabling the first and second flash memory chips to read packets of address information. At times t5, t6, and t7, the first and second flash signals (wave forms 902 and 904) each transmit first, second, and third address packets ADD0, ADD1, and ADD2 respectively to the first and second flash memory chips. At a time t8, the ALE signal (wave form 908) is deactivated, transitioning from the HIGH state to a LOW state, thereby disabling the first and second flash memory chips from reading address information. During time intervals between times t5 and t6, t6 and t7, and t7 and t8, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the read the first, second, and third address packets ADD0, ADD1, and ADD2 respectively. The three address packets ADD0, ADD1, and ADD2 specify a row-portion 732, 733 within a first sub-block 730 (Fig. 16).

At a time t9, the first and second flash signals (wave forms 902 and 904) begin simultaneously transmitting interleaved even and odd data bytes wherein the even and odd bytes form one sector of information. The even bytes are transmitted to the first flash memory chip via bus 680 (Fig. 6) and the odd sector bytes are transmitted to the second flash memory chip via bus 684 (Fig. 6). The even data bytes D0, D2, D4,....D510 of the even sector are received by the first flash chip and stored in the first even sector field 734 (Fig. 16) of the corresponding location 732 of the first flash memory chip. This is done by storing a byte each time the write enable signal WE* (Wave form 910) is activated. The odd data bytes D1, D3, D5,....D511 of the even sector are received by the second flash chip and stored in the second even sector field 742 (Fig. 16) of the corresponding location 733 thereof with each byte being stored when the WE* signal is activated. At a time t10, the first and second flash signals (wave forms 902 and 904) complete transmission of the interleaved even and odd data bytes of the even sector.

Immediately after time t10, during an interval between time t10 and a time t11, the first flash signal (wave form 902) transmits four packets of filler information (FFH, hexadecimal F, equivalent binary value "1111," decimal value "15") to the first flash memory chip via the first split bus 680 (Fig. 6) while the second flash signal (wave form 904) transmits error correction codes (ECC) to the second flash memory chip via the second split bus 684 (Fig. 6). The filler information FFH transmitted during this time period is received by the first flash memory chip and stored in the first spare field 736 (Fig. 16). The error correction code transmitted during this time period is received by the second flash memory chip and stored in the first error correction field 744 (Fig. 16) of the nonvolatile memory section 733 of the second flash memory chip. This error correction code, generated by ECC logic unit 660 (Fig. 16), relates to the even sector transmitted during the preceding time interval between time t10 and t11.

At a time t11, the first and second flash signals (wave forms 902 and 904) begin simultaneously transmitting interleaved even and odd data bytes, synchronous with the write enable signal WE* (wave form 910), of an odd sector to the first and second flash memory chips via the first and second first split buses 680 and 684 (Fig. 6) respectively. The even data bytes D0, D2, D4,....D510 of the odd sector are received by the first flash chip and stored to the first odd sector field 738 (Fig. 16) of the corresponding location 732 of the first flash memory chip. The odd data bytes D1, D3, D5,....D511 of the odd sector are received by the

second flash memory chip and stored to the second odd sector field 746 (Fig. 16) of the corresponding location 733 of the second flash memory chip. At a time t12, the first and second flash signals (wave forms 902 and 904) complete transmission of the interleaved even and odd data bytes of the odd sector.

Immediately after time t12, during an interval between time t12 and a time t13, the first flash signal (wave form 902) transmits no information to the first flash memory chip thereby maintaining the value in corresponding storage location bytes of the first flash memory chip at FFH (hexadecimal) or all 1's in binary. Meanwhile, between time t12 and time t13, while the second flash signal (wave form 904) transmits error correction codes (ECC) to the second flash memory chip via the second split bus 684 (Fig. 6). The filler information FFH transmitted during this time period is received by the first flash memory chip and stored to the second spare field 740 (Fig. 16). The error correction code transmitted during this time period is received by the second flash memory chip and stored to the second error correction field 748 (Fig. 16) of the nonvolatile memory section 733 of the second flash memory chip. This error correction code, generated by ECC logic unit 660 (Fig. 16), relates to the odd sector transmitted during the preceding time interval between time t11 and t12.

At a time t17, the first and second flash signals (wave forms 902 and 904) each transmit a read command signal 70H to the first and second first and second flash memory chips via the first and second split buses 680 and 684 respectively. While the read command signals 70H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the read command signals 70H. At a time t18, the CLE signal (wave form 906) is deactivated, transitioning back to the LOW state, thereby disabling the flash memory chips from reading command signals.

At a time t18, the first and second flash signals (wave forms 902 and 904) each transmit a status command signal STATUS to the first and second first and second flash memory chips via the first and second split buses 680 and 684 respectively. While the read command signals 70H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the read command signals 70H.

Fig. 10 shows a table diagram generally illustrating the memory storage format, as depicted in Fig. 7, for storing a block of information in memory bank 506 (Fig. 6) wherein a

single sector is written to a particular memory row location of the memory bank. As shown, a memory row location 728 designated ROW1 has an even sector S2 and an odd sector S3 stored therein in accordance with the format described above in reference to Fig. 7. A memory row location 728 designated ROW2 has a single even sector S4 stored in the first and second even sector fields 734 and 742 of a corresponding pair of row-portions of the first and second flash memory chips 670, 672. Because no odd sector is required to be stored in this case, fields 736, 738, 746, 748, 750, and 752 are shown to be erased.

Fig. 11 shows a table diagram illustrating the alternative memory storage format, as depicted in Fig. 7, for storing a block of information in memory bank 506 (Fig. 6) wherein a single sector is written to a particular memory row location of the memory bank. As mentioned above, field 764 is a three bit field which is used for storing the old row flag in the first bit place, the used/free row flag in the second bit place, and the defect row flag in the third bit place. Also as described above, field 766 is a two bit field which is used for storing the even sector move flag in the first bit place and the odd sector move flag in the second bit place.

The memory row location designated ROW1, having sectors S2 and S4 stored therein, has a value "00" stored in field 766 indicating that both sectors have been moved elsewhere in the memory bank. The memory row location designated ROW2, having a single even sector S4 stored in the first and second even sector fields 734 and 742, has a value "01" stored in field 766 indicating that the information in S4 has been updated and now resides elsewhere in the memory bank. A value of logic state "0" generally indicates that moved sectors have been updated by the host. Therefore, when the remaining sectors are moved from the old block which was not updated by the host, it can be determined that these sectors are not to be overwritten by the old data during the move.

A memory location 728 designated ROW1 has an even sector S2 and an odd sector S3 stored therein in accordance with the format described above in reference to Fig. 7. A memory location 728 designated ROW2 has a single even sector S4 stored in the first and second even sector fields 734 and 742 of a corresponding pair of row-portions of the first and second flash memory chips 670, 672. Because no odd sector is required to be stored in this case, fields 736, 738, 746, 748, 750, and 752 are shown to be erased.

Fig. 12 is a flowchart illustrating a process of simultaneously writing two sectors of information to two memory units during a single write operation in accordance with the

present invention. In step 1202, the memory controller 510 (Fig. 6) receives host addressing information from host 504 which specifies addresses for one or more sector locations, in the form of a logical block address (host-provided LBA) or in the form of cylinder head sector (CHS) information. If the host addressing information is in the form of CHS information, the controller translates the CHS information to LBA information. As mentioned, the sectors are organized in blocks and therefore, the host-provided LBA's may correspond to sectors of more than one block. This information is used by microprocessor 620 (Fig. 6) as will be further discussed below.

Microprocessor 620 (Fig. 6) executes instructions, which are stored in code storage unit 626 (Fig. 6) to carry out the depicted process. In step 1204, a sector count value is set equal to the number of sector locations of a current block, being addressed by the host wherein a sector location may, for example, be comprised of fields 734 and 742 (Fig. 7) or fields 738 and 746 (Fig. 7) of the memory bank. The microprocessor determines at 1206 whether or not each of the sector locations specified by the host-provided LBA values has been accessed by the host before. This determination is made by reading the contents of the corresponding virtual PBA field 804 (Fig. 8A) of the LBA-PBA map 800 stored in SPM RAM 720 (Fig. 6). As explained above in reference to Fig. 8A, if the virtual PBA value corresponding to a host-provided LBA is set to the all '1's state, then the corresponding LBA was not accessed by the host before. Memory space in memory bank 506 is erased a block at a time. If any sectors of a block have been accessed since a last erasure of the block, then the block is indicated as having been accessed by virtue of the virtual PBA value in field 804 (Fig. 8A) of the corresponding map row location of the LBA-PBA map being a value other than "all 1's".

If it is determined that one or more sector locations, of the current block, specified by the host-provided-LBA's have been accessed previously by the host, the write process proceeds to step 1210 in which microprocessor 620 (Fig. 6) sets the corresponding one of the move flags 760, 762 (Fig. 7) corresponding to the current sector location, and the write process proceeds to step 1208. As earlier discussed, maintaining the 'move' flag in non-volatile memory is optional and may be entirely eliminated without departing from the scope and spirit of the present invention. In the absence of move flags, the microprocessor maintains the status of sectors as to whether or not they have been moved to other blocks. This is done by keeping track of two values for each block. One value is the starting sector

location within a block where sectors have been moved and the second value is the number sectors within the block that have been moved. With these two values, status information as to whether or not and which sectors of a block have been moved to other block(s) may be reconstructed.

If it is determined, at step 1206, that none of the sector locations of the current block specified by the host-provided-LBA have been previously accessed, the write process proceeds directly to step 1208.

In step 1208, the space manager 636 (Fig. 6) of the controller searches for a free (or unused) block, such as block 727 (Fig. 7) located within the nonvolatile memory bank, each free block being identified by a specific virtual PBA value. The microprocessor determines at 1212 whether a free block is located, and if not, an error is reported by the controller 510 (Fig. 6) to the host indicating that the nonvolatile memory bank is unable to accommodate further storage of information. As this can result in a fatal system error, the inventors of the present invention have exercised great care in preventing this situation from occurring.

Once a free block within the nonvolatile memory is located at step 1208, the depicted process proceeds to step 1214. In step 1214, microprocessor 620 prompts space manager 636 (Fig. 6) to assign a virtual PBA value 822 (Fig. 8B) to the free block found in step 1208. This virtual PBA value is stored in the LBA-PBA map 800 (Fig. 8A) in a map row location 802 (Fig. 8A) identified by the masked bits 814 (Fig. 8B) of the host-provided LBA corresponding to the current block. The masked bits 814 (Fig. 8B) of the current host-provided LBA are obtained by shifting the host-provided LBA to the right by the 5 offset bits (or by dividing by 32). For example, if the host-identified LBA is 16H (hexadecimal notation), the row in which the virtual PBA is stored is row 0. Also at step 1214, the microprocessor appends the 'offset' bits 812 (Fig. 8B) to the virtual PBA corresponding to the found free block to obtain an actual PBA value 820 (Fig. 8B). At 1216, the microprocessor determines whether the actual PBA value is an even or odd value. At 1216, alternatively, the host-provided LBA may be checked in place of the actual PBA value to determine whether this value is odd or even.

If it is determined at 1216 that the actual PBA value is even, the process proceeds to 1218 at which the microprocessor determines whether the sector count is greater than one, i.e., there is more than one sector of information to be written at the point the controller requests that more than one sector to be transferred from the host to the internal buffer of the

controller and the process proceeds to 1232 at which the microprocessor determines whether two sectors of information have been transferred from the host to the data buffer 614 (Fig. 6) (through the host interface circuit 610). That is, where there is more than one sector of information that needs to be written to nonvolatile memory, as detected by the flash state machine 642, two sectors of information are transferred at-a-time from the host to the data buffer 614. The data buffer 614 is used to temporarily store the sectors' information until the same is stored into the memory bank 506. In the preferred embodiment, each sector includes 512 bytes of user data and 16 bytes of overhead information.

Where two sectors of information have not yet been transferred to the data buffer 614, the microprocessor waits until such a transfer is completed, as shown by the 'NO' branch loop at 1232.

At step 1234, the microprocessor initiates the writing of the two sectors that have been temporarily saved to the data buffer to the memory bank 506 (Fig. 6) by issuing a write command, followed by address and data information. The write operation at step 1234 is performed according to the method and apparatus discussed above relative to Figs. 7 and 9.

Upon completion of writing two sectors of information, the write operation is verified at 1235. If information was not correctly programmed into the sectors at step 1234, the process continues to step 1237 where a defect management routine is performed, as will be discussed in greater detail below. After execution of the defect management routine, the sector count is decremented by two at step 1236. At 1235, if the write operation was verified as being successful, step 1236 is executed and no defect management is necessary. The microprocessor then determines at 1238 whether the sector count is equal to zero and if so, it is assumed that no more sectors remain to be written and the process proceeds to 1228. If, however, more sectors need to be written the process proceeds to step 1240 at which the host-provided LBA is incremented by two to point to the next sector that is to be written.

At step 1240, the microprocessor determines whether the last sector of the block has been reached. The block boundary is determined by comparing the 'offset' value of the current LBA to the number of sectors in a block, and if those values are equal, a block boundary is reached. For example, in the preferred embodiment, since a block includes 32 sectors, the 'offset' value of the current LBA is compared against '32' (in decimal notation). If alternatively, a block is defined to have other than 32 sectors, such as 16 sectors, the latter is compared against the 'offset'. If a block boundary in the nonvolatile memory is reached,

the write process continues from step 1206 where the virtual PBA value corresponding to the current LBA value is checked for an all '1's condition and so on. If a block boundary is not reached at step 1242, the write process continues from step 1218.

At step 1218, if it is determined that the sector count is not greater than one, the microprocessor proceeds to determine at 1220 whether data buffer 614 (Fig. 6) has received at least one sector of information from the host. If not, the microprocessor waits until one sector of information is transferred from the host to the data buffer 614. Upon receipt of one sector of information, writing of the next sector is initiated and performed at step 1222 according to the method and apparatus discussed above relative to Figs. 10 and 11. Upon completion of writing a sector of information, the write operation is verified at 1223. If information was not correctly programmed into the sector at step 1222, the process continues to step 1225 where a defect management routine is performed, as will be discussed in greater detail below. After execution of the defect management routine, at step 1224, the sector count is decremented by one. If at 1223, it is determined that the write operation was correctly performed, the process continues to step 1224 and no defect management routine is executed. At 1226, the microprocessor determines whether the sector count is equal to zero and, if not, the host-provided LBA is incremented by one and the write process continues to step 1242 where the microprocessor checks for a block boundary as explained above.

If at step 1226, as in step 1238, it is determined that no more sectors remain to be written, i.e. the sector count is zero, the depicted process proceeds to 1228 at which the microprocessor determines whether the move flag is set. As noted above, the move flag would be set at step 1210 if it was determined at 1206 that an LBA was being re-accessed by the host.

If it is determined at 1228 that the move flag is not set, the write process ends. However, upon a determined at 1228 that the move flag is set, the block is updated. That is, those sectors of the current block that were not accessed are moved to corresponding sector locations in the block within memory bank 506 identified by the virtual PBA value assigned in step 1214 to the free block found in step 1208. This is perhaps best understood by an example.

Let us assume for the purpose of discussion that the sectors identified by LBAs 1, 2, 3, 4, 5 and 6 have already been written and that the host now commands the controller to write data to sectors identified by LBAs 3, 4 and 5. Further, let us assume that during the first write

process when LBAs 1-6 were written, they were stored in a block location in the memory bank 506 (Fig. 6) identified by a virtual PBA value of "3" and the LBA locations 3, 4 and 5 are now (during the second write process) being written to a location in the memory bank identified by a virtual PBA value of "8". During writing of locations identified by host-provided LBA values of 3, 4, and 5, the microprocessor at step 1206 determines that these block locations are being re-accessed and the move flag at 1210 is set. Furthermore, at step 1230, after the sectors, identified by host-provided LBAs 3, 4, and 5, have been written to corresponding sectors of the block identified by virtual PBA "8", sectors in the block identified by virtual PBA "3" that were not re-accessed during the write operation are moved from the block identified by virtual PBA "3" to corresponding sector locations of the block identified by virtual PBA "8" and the block identified by virtual PBA "3" is thereafter erased. This example assumes that remaining sectors of the block identified by virtual PBA "3", such as sectors 0 and 7-31 (assuming there are 32 sectors in a block), were not accessed since the last erase of the block in which they reside and therefore contain no valid sector information. Otherwise, if those sectors were previously accessed, then they would also be moved to the virtual PBA location 8.

Step 1230 may be implemented in many ways. The inventors of the present invention disclose various methods and apparatus which may be alternatively employed for performing the move operation of step 1230. In patent applications, Serial No. 08/946,331 entitled "Moving Sequential Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", filed on Oct. 7, 1997, and Serial No. 08/831,266 entitled "Moving Sectors Within a Block of Information In a Flash Memory Mass Storage Architecture", filed on March 31, 1997, the disclosures of which are herein incorporated by reference.

Fig. 12a shows the steps performed by the micorprocessor if the defect management routine at steps 1237 and 1225 (in Fig. 12) is executed. The block management routine is executed when the write operation is not successfully verified; the block(s) being programmed is in some way defective and a different area in the nonvolatile memory, i.e. another block need be located for programming therein.

At step 1600, the block that was being unsuccessfully programmed is marked as "defective" by setting the "defect" flags 756 (in Fig. 7). At step 1602, the space manager within the controller is commanded to find a free block. At step 1604, the information that would have been programmed at steps 1234 and 1222 (in Fig. 12) i.e. the block marked

"defective" is programmed into corresponding sector locations within the free block found in step 1602.

At step 1606, the block marked "defective" is checked for the presence of any sector information that was previously written thereto successfully. If any such sectors exist, at step 1608, these previouly-programmed sectors are moved to the free block, as is additional block information in the process of Fig. 12.

Fig. 13 shows a table diagram generally illustrating a memory storage format for storing a block, including 32 sectors, of information in memory bank 506 in accordance with an alternative embodiment of the present invention. In this embodiment, an even sector is stored in a first row portion located in a first of the two memory units and an odd sector is stored in a second row portion located in the second of the two memory units. In the depicted embodiment, memory bank 506 includes a plurality of M blocks 1302 designated BLCK0, BLCK1, BLCK(M-1) each having a physical block addresses (PBA). Each of the blocks 1302 includes a plurality of N memory row locations 1304, and in a preferred embodiment, N=16. Each block 1302 of memory bank 506 is comprised of a first sub-block 1306 of first flash memory chip 670, and a corresponding second sub-block 1308 of second flash memory chip 672 wherein the corresponding sub-blocks are identified by the same virtual PBA. Each memory row location 1304 includes a first row-portion 1310 and a corresponding second row-portion 1312. In the depicted embodiment each of the first and second row-portions 1310, 1312 includes storage for 512 bytes of data information plus additional storage space for error correction information (ECC information) and flag information.

Each of the first row-portions 1310 includes an even sector field 1314 for storing an even sector (S0, S2, S4,...) of information, and an even sector error correction field 1316 for storing error correction information corresponding to the even sector stored in field 1314. Each of the second row-portions 1312 includes an odd sector field 1318 for storing an odd sector (S1, S3, S5,...) of information, an odd sector error correction field 1320 for storing error correction information corresponding to the odd sector stored in 1318, a block address field 1322, and a flag field 1324. It is understood in the present invention that field 1314 could alternatively be used to store an odd sector while field 1318 could alternatively be used to store an even sector. Also, first row-portion 1310 could alternatively be used for storing the block address and flags.

Flag field 1324 is used for storing flag information which is used by controller 510 (Fig. 6) during access operations as further explained below. Block address field 1322 is used for storing the block address permanently assigned to block 1302, such as "0" for BLCK0. Only a single block address entry is required in the block address field per block. In a preferred embodiment, a block address entry is entered in block address field 1322 of the last row 1304, which is row 15.

In this alternative embodiment, the first and second split buses 680, 684 (Fig. 6) include lines coupled to receive data bytes of the even and odd sectors respectively. The controller 510 (Fig. 6) writes two sectors simultaneously by simultaneously writing a byte of an even sector and an odd sector simultaneously via the first and second split buses 680, 684 (Fig. 6), respectively. The split buses 680, 684 (Fig. 6) also provide for: transmission of ECC information between the flash memory chips and the flash state machine 642 and ECC logic unit 660 of the memory controller 510; and transmission of address information from flash state machine 642 to the flash memory chips.

Fig. 14 shows a timing diagram illustrating the timing of control signals, address signals, and data signals for an erase operation of the memory system of Fig. 6. The diagram includes: the wave form 902 representing the first flash signal which transmits time multiplexed command, address, and data information from the flash state machine 642 (Fig. 6) via first split bus 680 (Fig. 6) to the first flash memory chip; the wave form 904 representing the second flash signal which transmits time multiplexed command, address, and data signals transmitted from the flash state machine via second split bus 684 (Fig. 6) to the second flash memory chip; a time line 1450; and a plurality of control signal wave forms. The control signal wave forms, all of which are described above, include: wave form 906 representing the command line enable (CLE) signal; wave form 908 representing the address line enable (ALE) signal; wave form 910 representing the write enable (WE) signal; wave form 912 representing the read enable (RE) signal; wave form 914 representing the flash chip enable (FCE*) signal; and wave form 916 representing the flash ready/busy signal (FRDY_BSY* signal).

The erase operation commences at a time E0 at which the FCE* signal (wave form 914) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to begin receiving command, address, and data signals. At a subsequent time E1, the CLE signal (wave form 906) is activated, transitioning from a LOW state to a HIGH

state, thereby enabling the first and second flash memory chips to read command signals. At a time E2, the first and second flash signals (wave forms 902 and 904) each transmit a command signal. The first flash signal (wave form 902) transmits an 'erase set' command, 60H, via the first split bus 680 (Fig. 6) to the first flash memory chip while the second flash signal (wave form 904) transmits a read status command signal 70H via the second split bus 684 (Fig. 6) to the second flash memory chip. At a time E3, while the command signals 60H and 70H are active, the WE signal (wave form 910) transitions from a HIGH state to a LOW state thereby enabling the first and second flash memory chips to read the command signals 60H and 70H. At a time E4, the CLE signal (wave form 906) is deactivated, transitioning back to the LOW state, thereby disabling the flash memory chips from reading command signals.

Also at time E4, the ALE signal (wave form 908) is activated, transitioning from a LOW state to a HIGH state, thereby enabling the first and second flash memory chips to read packets of address information. At times E5 and E6, the first flash signal (wave form 902) transmits first and second address packets ADD0 and ADD1 respectively to the first flash memory chip wherein the first and second address packets ADD0 and ADD1 specify a sub-block 730 (Fig. 7) of the first flash memory chip 670 of the memory bank. At a time E7, the ALE signal (wave form 908) is deactivated. During time intervals between times E3 and E4, and E4 and E5, the WE signal (wave form 910) transitions to the LOW state to enable the flash memory chip to read the address packets.

At a time E8, the CLE signal (wave form 906) is again activated to enable the first and second memory chips to read command signals. At a time E9, the first flash signal (wave form 902) transmits DOH, which is an 'erase confirm command' to the first flash memory chip. This command, as sampled by the CLE signal, actually initiates the erase operation within the flash chips, after which, the contents of data fields 734 and 738 of each memory row portion 732 of the addressed sub-block 730 (Fig. 7) of the first flash memory chip 670 are erased, i.e. set to an "all 1's" state. At a time E10, the FRDY-BSY* signal (wave form 912) transitions from a HIGH state to a LOW state to indicate to the flash state machine 642 (Fig. 6) that at least one of the flash memory chips is busy.

At a time E11, the CLE signal (wave form 906) is activated to enable the first and second flash memory chips to read command signals. At a time E12, the first and second flash signals (wave forms 902 and 904) each transmit a command signal. The first flash

signal (wave form 902) transmits a read command signal 70H via the first split bus 680 (Fig. 6) to the first flash memory chip while the second flash signal (wave form 904) transmits an erase command signal 60H via the second split bus 684 (Fig. 6) to the second flash memory chip. At a time E13, while the command signals 70H and 60H are active, the WE signal (wave form 910) transitions to the LOW state to enable the first and second flash memory chips to read the command signals 60H and 70H. At a time E14, the CLE signal (wave form 906) is deactivated to disable the flash memory chips from reading command signals and the ALE signal (wave form 908) is activated thereby enabling the first and second flash memory chips to read packets of address information. At times E15 and E16, the second flash signal (wave form 904) transmits first and second address packets ADD0 and ADD1 respectively to the second flash memory chip wherein the first and second address packets ADD0 and ADD1 specify a sub-block 731 (Fig. 7) of the second flash memory chip 672 of the memory bank. At a time E17, the ALE signal (wave form 908) is deactivated. During time intervals between times E13 and E14, and E14 and E15, the WE signal (wave form 910) enables the flash memory chips to read the address packets. At a time E18, the CLE signal (wave form 906) is again activated to enable the first and second memory chips to read command signals. At a time E19, the first flash signal (wave form 902) transmits DOH to the first flash memory chip to erase the contents of data fields 734 and 738 of each memory row portion 732 of the specified block and thereby set them to an "all 1's" state.

To summarize, during a time interval TEB1, between the times E0 and E11, the memory controller erases an addressed sub-block 730 (Fig. 7) of the first flash memory chip 670. Also, during a time interval TEB2, between the times E11 and E20, the memory controller erases a corresponding addressed sub-block 731 (Fig. 7) of the second flash memory chip 672. At a time E21, the FRDY_BSY* signal (wave form 916) transitions from a LOW state to a HIGH state to indicate to the flash state machine 642 (Fig. 6) that both of the flash memory chips are finished with the erase operation.

Immediately after time E21, the first and second flash signals (wave forms 902 and 904) each transmit a read status command signal 70H to the first and second flash memory chips respectively. While the read command signals 70H are active, the WE signal (wave form 910) transitions to the LOW state thereby enabling the first and second flash memory chips to read the read command signals 70H. At a time E22, the first and second flash signals (wave forms 902 and 904) both transmit a status data back to the controller.

So, the status of both flash memory chips are read simultaneously after the erase operation is performed on the two corresponding addressed sub-blocks of the flash memory chips as described above.

If either of the sub-blocks 730, 731 of the memory chips has an error, the entire block 727 (Fig. 7) within the chips is marked defective by setting the contents of the defect flag 756 (Fig. 7) in the second flash memory chip 672.

Fig. 15 is a flowchart illustrating a process of erasing a block, including a first sub-block stored in a first memory unit and a second sub-block stored in a second memory unit, in accordance with the present invention. Microprocessor 620 (Fig. 6) executes instructions, which are stored in code RAM 626 (Fig. 6) to carry out the depicted process.

In step 1502, microprocessor 620 (Fig. 6) loads a block address to be erased. In step 1504, the microprocessor initiates the erase operations described above in reference to the timing diagram at 1400 (Fig. 14). At 1506, the microprocessor determines whether the erase operation is finished by reading the flash ready/busy (FRDY_BSY*) signal (wave form 916 of Fig. 14) which transitions from a LOW state to a HIGH state to indicate to the flash state machine 642 (Fig. 6) that both of the flash memory chips are finished with the erase operation. At 1508, the microprocessor reads the status of the flash chips 670, 672 (Fig. 6). At 1508, the microprocessor determines whether the erase operation performed in step 1504 was successful in both of the flash chips 670, 672 (Fig. 6) and, if so, the process ends. If it is determined that the erase operation performed in step 1504 was not successful in both of the flash chips, then the microprocessor marks the block in both of the flash chips 670, 672 defective.

Although the present invention has been described in terms of specific embodiments, it is anticipated that alterations and modifications thereof will no doubt become apparent to those skilled in the art. It is therefore intended that the following claims be interpreted as covering all such alterations and modification as fall within the true spirit and scope of the invention.

What is claimed is:

CLAIMS

1. A memory storage device for storing information organized in sectors within a nonvolatile memory bank, each said sector including a user data portion and an overhead portion, said sectors being organized into blocks, each sector identified by a host-provided logical block address (LBA) and an actual physical block address (PBA) derived from a virtual PBA, each block being identified by a modified LBA derived from said host-provided LBA and said virtual PBA, said host-provided LBA being received by said storage device from the host for identifying a sector of information to be accessed, said actual PBA developed by said storage device for identifying a free location within said memory bank wherein said accessed sector is to be stored, said storage device comprising:

a memory controller coupled to said host; and

a nonvolatile memory bank coupled to said memory controller via a memory bus, said memory bank including a first non-volatile semiconductor memory unit and a second non-volatile semiconductor memory unit, said memory bank having storage blocks each of which includes at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, each said memory row location providing storage space for two of said sectors,

wherein said memory controller accesses two sectors of information simultaneously.

2. A memory storage device as recited in claim 1 wherein said memory controller includes:

a data buffer for temporarily storing said sector-organized information;

a microprocessor;

a space manager including a space manager controller and a space manager memory unit for maintaining a map for translating said LBA to said PBA; and

an error correction code logic unit for performing error coding and correction operations on said sector-organized information.

3. A memory storage device as recited in claim 1 wherein:

each of said first row-portions includes,

a first even sector field for storing even data bytes of an even sector, and
a first odd sector field for storing even data bytes of an odd sector;
each of said second row-portions includes,
a second even sector field for storing odd data bytes of said even sector, and
a second odd sector field for storing odd data bytes of said odd sector;
said memory bus includes,
a first split bus coupled to transmit said even data bytes of said sectors between said memory controller and said first memory unit;
a second split bus coupled to transmit said odd data bytes of said sectors between said memory controller and said second memory unit.

4. A memory storage device as recited in claim 3 wherein each of said second row-portions further includes:
a first error correction field for storing error correction information corresponding to said even sector;
a second error correction field for storing error correction information corresponding to said odd sector;
a block address field for storing said PBA which specifies an address of said corresponding block, and
a flag field for storing information indicative of the status of said corresponding block.

5. A memory storage device as recited in claim 4 wherein each of said first row-portions further includes:
a first error correction field for storing error correction information corresponding to said even sector;
a second error correction field for storing error correction information corresponding to said odd sector; and
a block address field for storing said PBA which specifies an address of said corresponding block, and
a flag field for storing information indicative of the status of said corresponding block.

6. A memory storage device as recited in claim 5 wherein:

each of said first and second row-portions includes storage space for 512 bytes of said user data plus an additional 16 bytes of storage space for said overhead information; and

said first split bus and said second split bus each include 8 bit lines.

7. A memory storage device as recited in claim 6 wherein said controller includes:

means for accessing an even sector of information by simultaneously accessing said first and second even sector fields of corresponding row-portions of said first and second memory units via said first and second split buses; and

means for accessing an odd sector of information by simultaneously accessing said first and second odd sector fields of corresponding row-portions of said first and second memory units via said first and second split buses.

8. A memory storage device as recited in claim 7 wherein said controller includes error correction code logic for performing error coding and correction operations on said sector-organized information.

9. A memory storage device as recited in claim 8 wherein said flag field is used to store block level flags including:

a used/free block flag indicating whether said corresponding block is currently being used to store information; and

a defect block flag indicating whether said corresponding block is defective

10. A memory storage device as recited in claim 9 wherein each of said corresponding sub-blocks are identified by a single PBA value.

11. A memory storage device as recited in claim 10 wherein said non-volatile memory units are flash memory chips.

12. A memory storage device as recited in claim 11 wherein:

each of said first row-portions includes a first sector field for storing data bytes of a first sector;

each of said second row-portions includes a second sector field for storing data bytes of a second sector;

said memory bus includes,

a first split bus coupled to transmit (least significant ?) data bytes of said sectors between said memory controller and said first memory unit;

a second split bus coupled to transmit (most significant ?) data bytes of said sectors between said memory controller and said second memory unit.

13. A memory storage device as recited in claim 12 wherein:

means for simultaneously accessing said first and second sector fields of corresponding row-portions of said first and second memory units via said first and second split buses

14. In a storage device including a non-volatile memory bank and a controller coupled to said memory bank via a memory bus, said memory bank including a first non-volatile memory unit and a second non-volatile memory unit, said memory bank having storage locations defined by blocks, each block for storing a sectors of information having a user data portion and an overhead portion, each block having associated therewith a modified logical block address (LBA) derived from a host-provided LBA value and an actual physical block address (PBA) derived from a virtual PBA value, said host-provided LBA value received by said controller from the host for identifying a sector of information to be accessed, said actual PBA developed by said storage device for identifying a free location within said memory bank wherein a sector of information identified by the host is to be stored, each block including at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, a process of writing a first sector and a second sector to said memory bank during a single write operation, said process including the steps of:

simultaneously providing a write command to said first and second memory units;

addressing one of said memory row locations of said memory bank by simultaneously addressing corresponding first and second row portions of said addressed row location;

simultaneously providing a first data byte of said first and second sectors to said first memory unit, and a second data byte of said first and second sectors to said second memory unit.

15. In a storage device including a memory bank and a controller as recited in claim 14 wherein said first byte is an even data byte of one of said first and second sectors and said second data byte is odd data byte of one of said first and second sectors.

16. In a storage device including a memory bank and a controller as recited in claim 15 wherein said first data byte is a data byte of said first sector and said second data byte is a data byte of said second sector.

17. In a storage device including a memory bank and a controller as recited in claim 16 wherein an even sector and an odd sector are stored in a single memory row location and wherein even user data bytes of said first and second sectors are stored in said first memory unit and odd user data bytes of said first and second sectors are stored in said second memory unit and wherein overhead information associated with said first and second sectors is stored in one of said first and second row portions.

18. In a storage device including a memory bank and a controller as recited in claim 17 wherein a block address entry is entered in block address field in a last of said row locations of each said block.

19. In a storage device including a memory bank and a controller as recited in claim 18 wherein said controller accesses an even sector of information stored collectively in said first and second flash memory chips by simultaneously accessing first and second even sector fields of corresponding row-portions of said first and second flash memory chips via said first and second split buses

said first and second split buses including lines coupled to receive said even and odd data bytes respectively of a sector of information

said controller accesses an odd sector of information stored collectively in said first and second flash memory chips by simultaneously accessing said first and second odd sector fields , via said first and second split buses

said split buses also providing for transmission of ECC information between said flash memory chips and said flash state machine and ECC logic unit of said memory controller ; and transmission of address information from flash state machine to said flash memory chips.

20. In a storage device including a memory bank and a controller as recited in claim 19 wherein said flag field is used to store block level flags including:
a used/free block flag indicating whether said corresponding block is currently being used to store information
a defect block flag indicating whether said corresponding block is defective

21. In a storage device including a memory bank and a controller as recited in claim 20 wherein each of said corresponding sub-blocks are identified by a single virtual PBA value.

22. In a storage device including a nonvolatile memory bank and a controller coupled to said memory bank via a memory bus, said memory bank including a first non-volatile memory unit and a second non-volatile memory unit, said memory bank having storage blocks for storing information sectors each of which includes a user data portion and an overhead portion, each block having associated therewith a logical block address (LBA) and a physical block address (PBA), said LBA provided by a host to said controller for identifying a block to be accessed, said PBA developed by said storage device for identifying a free location within said memory bank wherein said accessed block is to be stored, each block including at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, a process of writing a first sector and a second sector to said memory bank during a single write operation, said process including the steps of:

simultaneously providing a write command to said first and second memory units;

addressing one of said memory row locations of said memory bank by simultaneously addressing corresponding first and second row portions of said addressed row location;

storing even data bytes of an even sector to a first even sector field of said first row-portion;

storing even data bytes of an odd sector to a first odd sector field of said first addressed row-portion;

storing odd data bytes of said even sector to a second even sector field of said second addressed row-portion; and

storing odd data bytes of said odd sector of information to a second odd sector field of said second addressed row-portion.

23. In a storage device including a memory bank and a controller as recited in claim 22 further including the steps of:

determining even sector error correction information corresponding to said even sector of information and odd sector error correction information corresponding to said odd sector of information;

storing said even sector error correction information to a first error correction field of said second addressed row-portion; and

storing said odd sector error correction information to a second error correction field of said second addressed row-portion.

24. In a storage device including a nonvolatile memory bank and a controller coupled to said memory bank via a memory bus, said memory bank including a first non-volatile memory unit and a second non-volatile memory unit, said memory bank having blocks for storing sectors of information each of which includes a user data portion and an overhead portion, each said block having associated therewith a logical block address (LBA) and a physical block address (PBA), said LBA provided by a host to said controller for identifying a sector of information to be accessed, said PBA developed by said storage device for identifying a free block location within said memory bank wherein said accessed block is to be stored, each block including at least one memory row location having a first row-portion located in said first memory unit, and a corresponding second row-portion located in said second memory unit, said controller including a data buffer; a process of writing sector-organized information to said memory bank, said process including the steps of:

receiving host-provided LBA values from said host, each said host-provided LBA value for identifying a sector of information;

modifying a current host-provided LBA to identify a block of sectors of information;

providing a map having map row locations identified by said modified LBA values or virtual PBA values, said map used as a look-up-table for storing virtual PBA values corresponding to modified LBA values;

setting a sector count value equal to the number of sectors of information identified by the host;

searching for a free block within said memory bank identified by a current virtual PBA;

storing said current virtual PBA, corresponding to said free block, in a map row location identified by said modified current LBA in said map;

determining whether said current host-provided LBA is even and whether said sector count is greater than one;

if said current actual PBA value is even and said sector count is greater than one,

simultaneously writing two sectors of information, one sector identified by said current host-provided LBA and a second sector of information identified by said current host-provided LBA plus one, said first sector being written to said first non-volatile memory unit and said second sector being written to second non-volatile memory units of said memory bank,

decrementing said sector count by two,

determining whether said sector count is equal to zero,

if said sector count is not equal to zero, increasing said current host-provided LBA value by two to point to the next sector that is to be written,

if said current host-provided LBA value or said sector count is not greater than one,

simultaneously writing even data bytes of a current sector of information identified by said current host-provided LBA to said first non-volatile memory unit and odd data bytes of said current sector to said second non-volatile memory unit,

decrementing said sector count by one,

determining whether said sector count is equal to zero, and

if said sector count is not equal to zero, increasing said current host-provided LBA by one to point to the next sector of information that is to be written.

25. In a storage device including a memory bank and a controller as recited in claim 24, wherein said step of simultaneously writing said two current sectors to said first and second non-volatile memory units of said memory bank, includes the steps of:

simultaneously providing a write command to said first and second memory units;

addressing one of said memory row locations of said memory bank by simultaneously addressing corresponding first and second row portions of said addressed row location;

storing even data bytes of an even sector to a first even sector field of said first addressed row-portion;

storing even data bytes of an odd sector to a first odd sector field of said first addressed row-portion;

storing odd data bytes of said even sector to a second even sector field of said second addressed row-portion; and

storing odd data bytes of said odd sector of information to a second odd sector field of said second addressed row-portion.

26. In a storage device including a memory bank and a controller as recited in claim 25, further including the steps of:

determining whether or not each of said addressed sector locations of a current block has been accessed since the last erasure thereof; (by reading the contents of the corresponding virtual PBA field of the LBA-PBA map);

setting a move flag corresponding to said current block if said current block has been accessed since the last erasure thereof;

subsequent to performing a write operation

determining whether said move flag is set;

if said move flag is set, updating said current block by moving those of said sectors not written to which belong to said current block to corresponding sector locations in said free block.

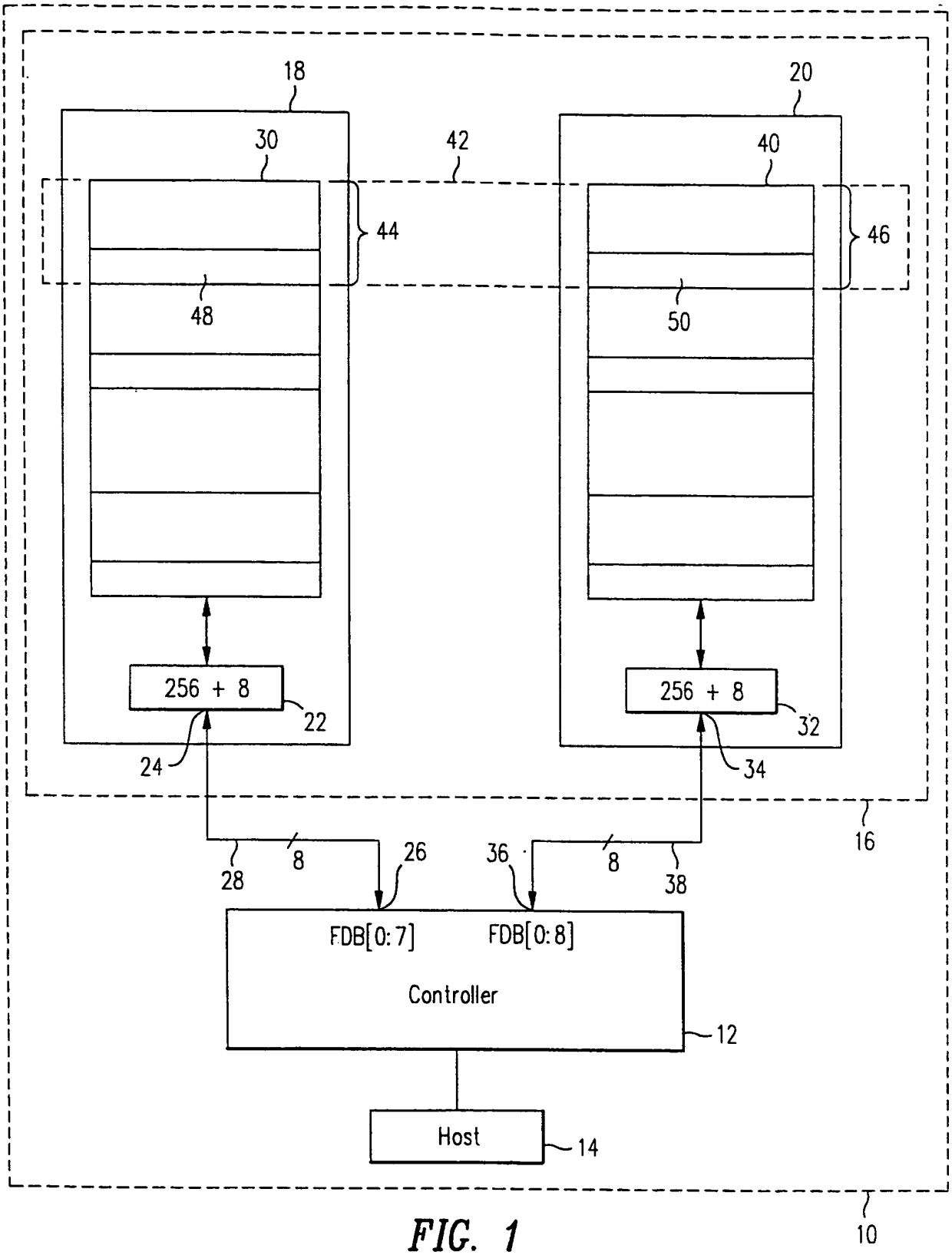


FIG. 1
(Prior Art)

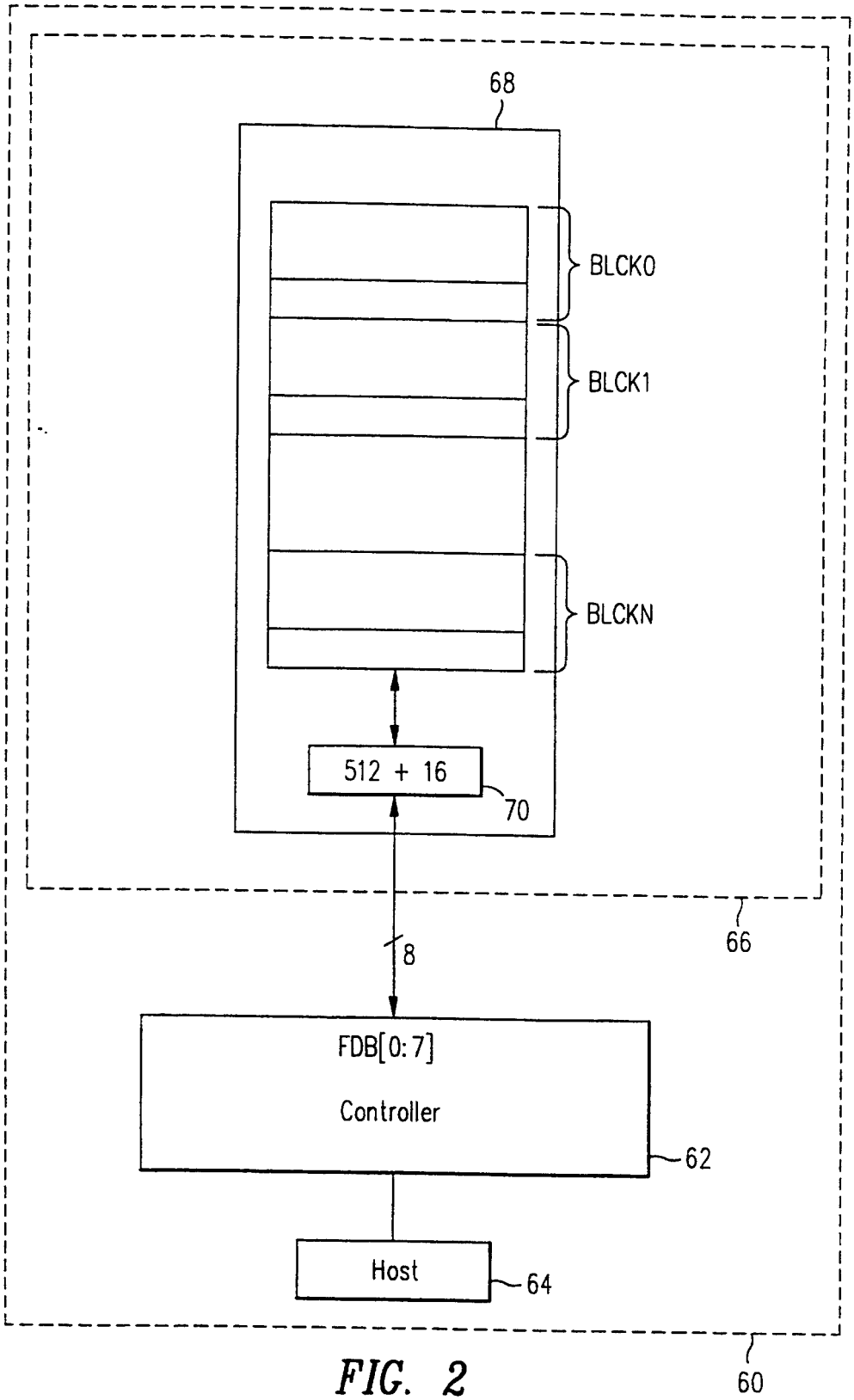


FIG. 2
(Prior Art)

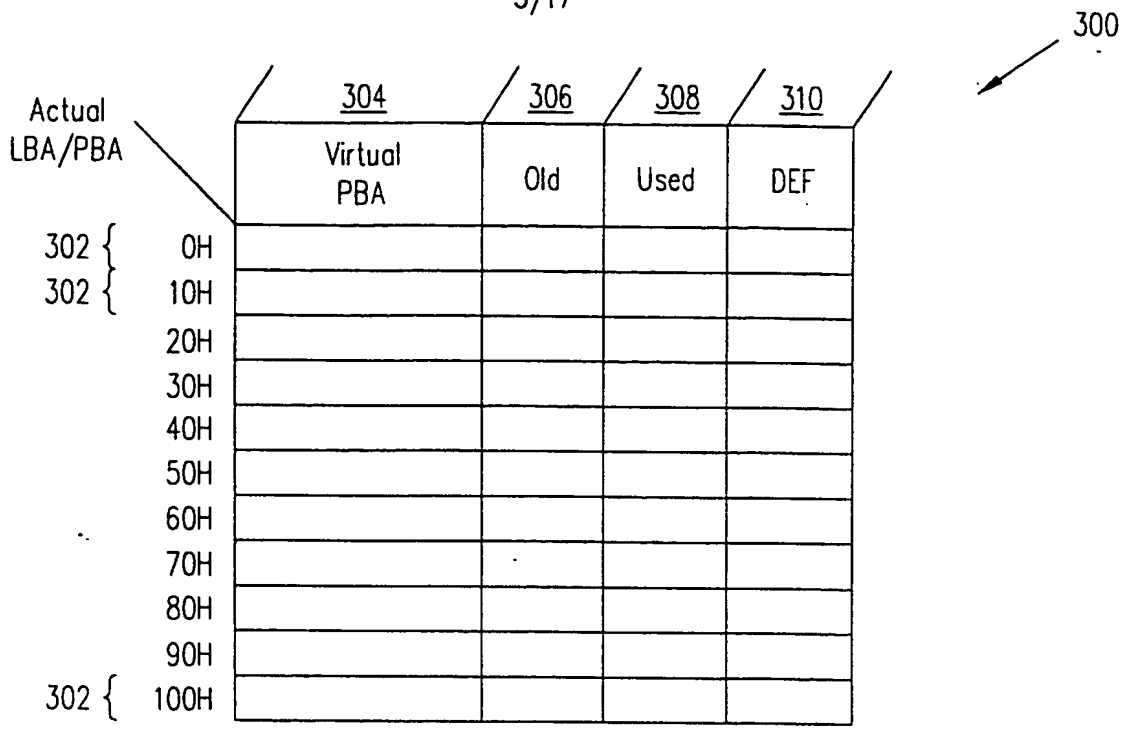


FIG. 3

(Prior Art)

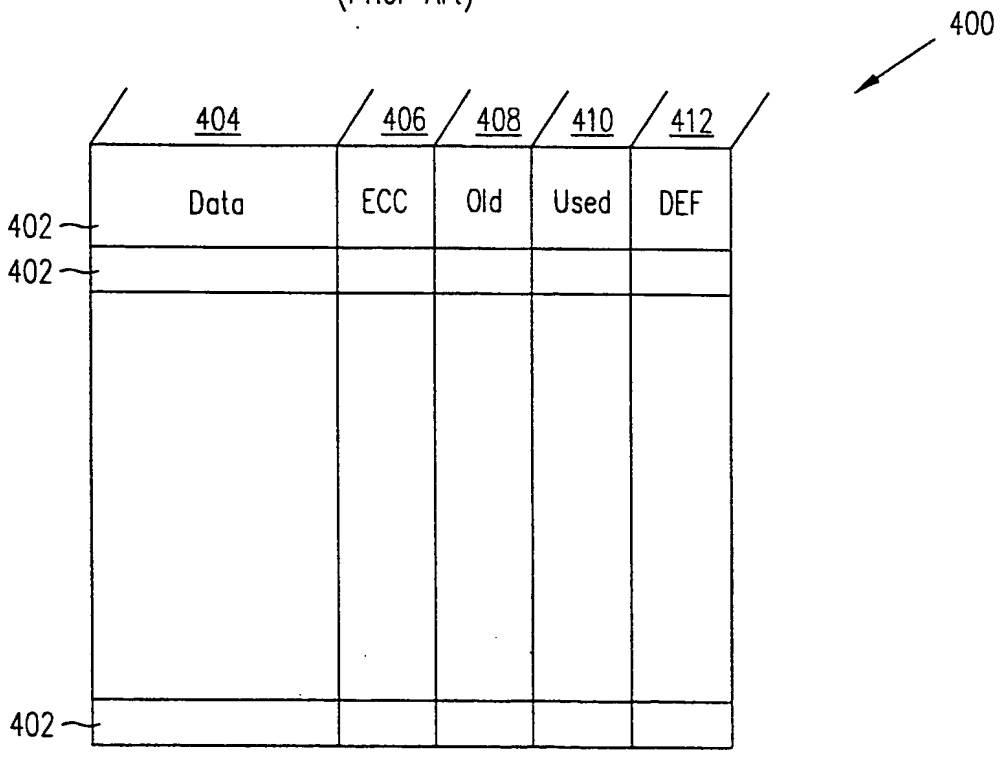


FIG. 4

(Prior Art)

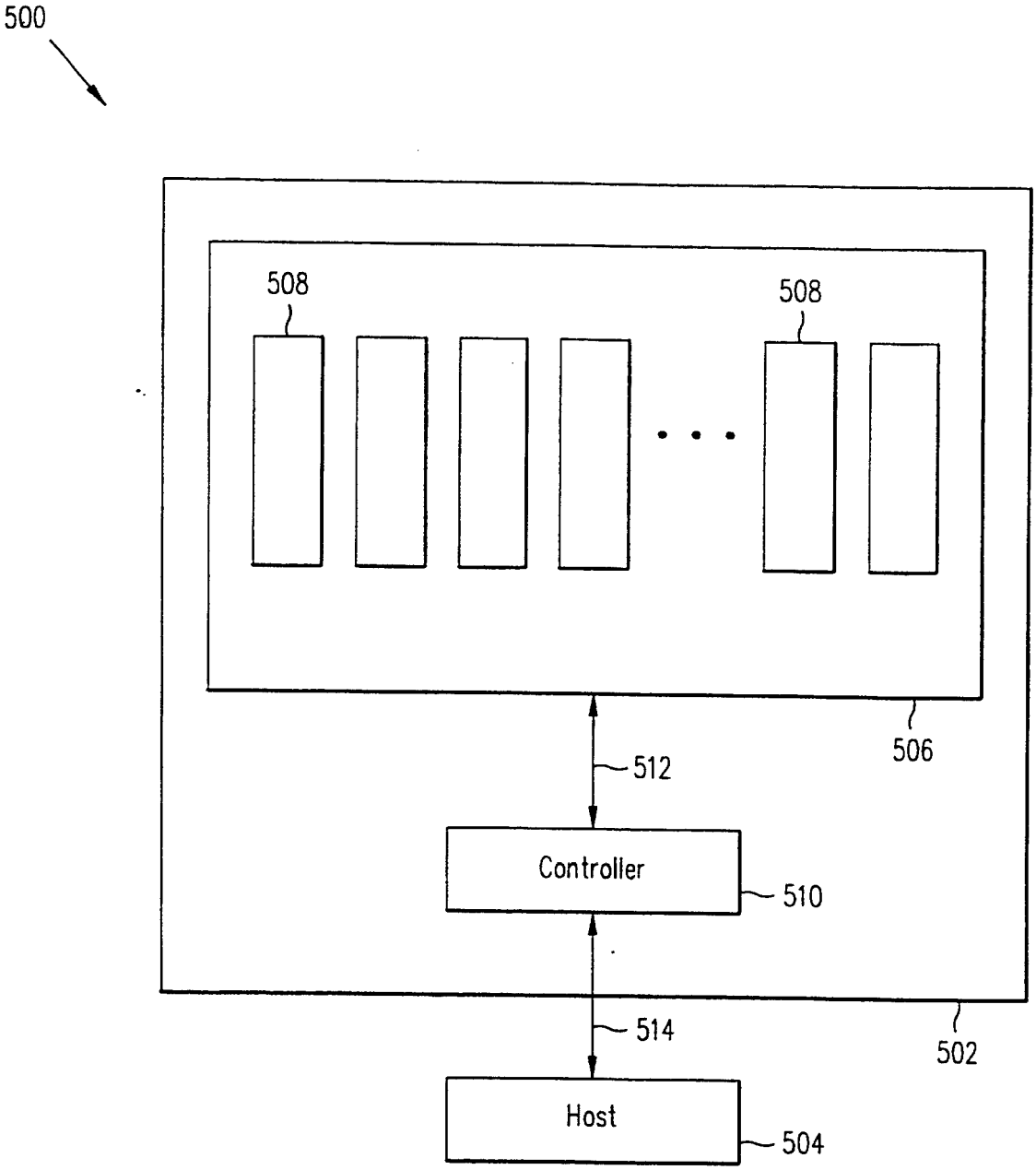
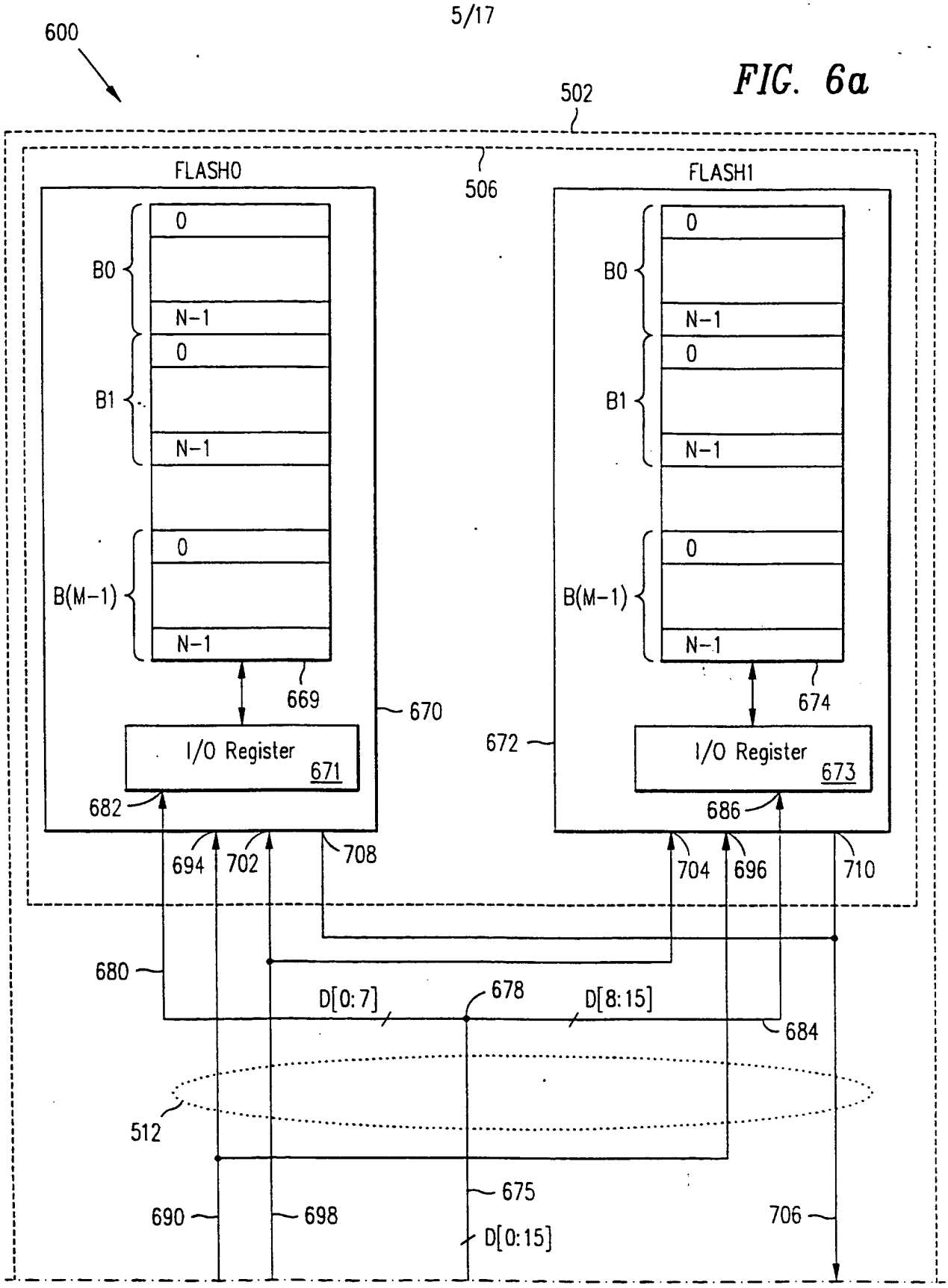


FIG. 5

FIG. 6a



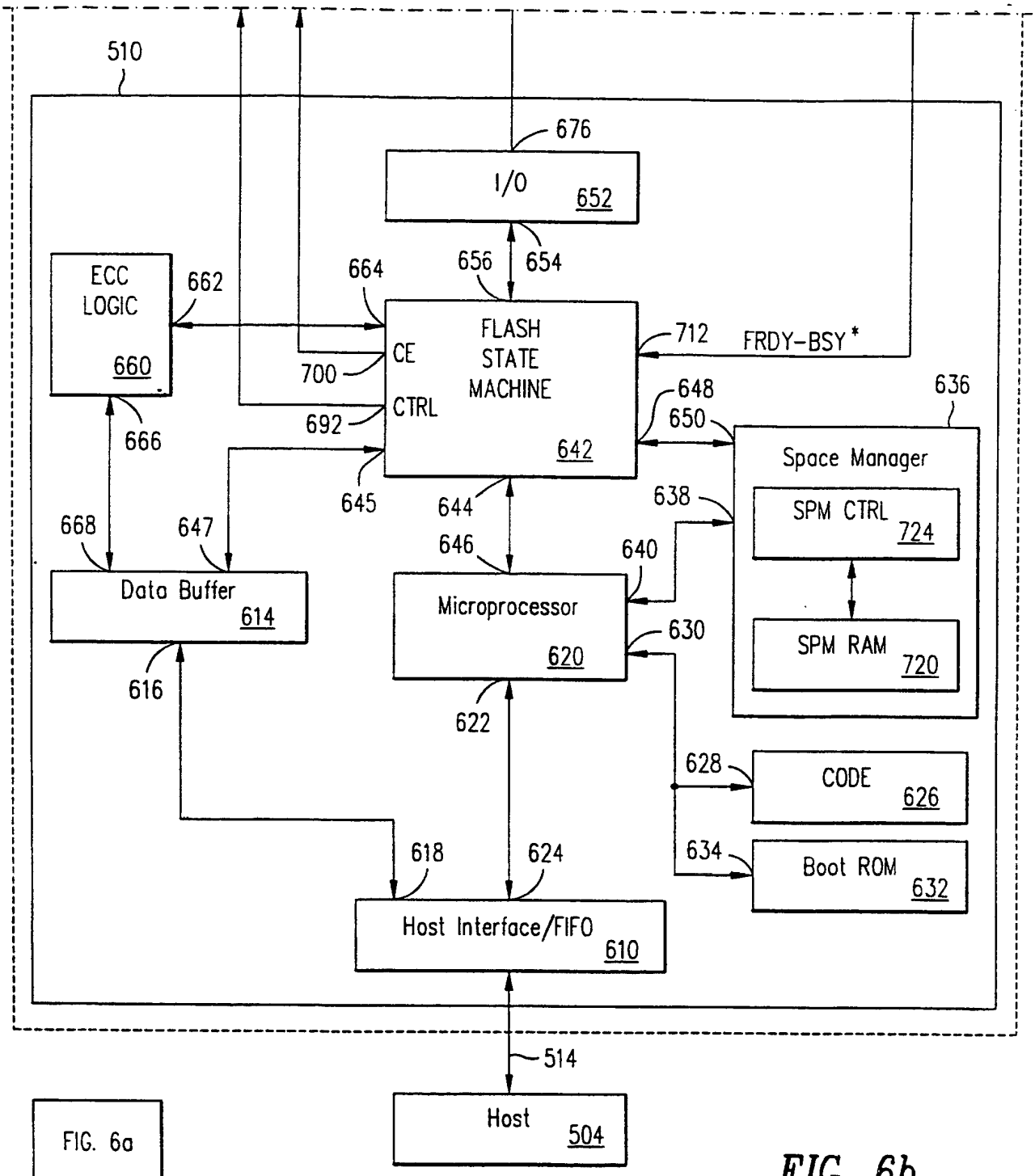


FIG. 6b

FIG. 6a

FIG. 6b

Key To

FIG. 6

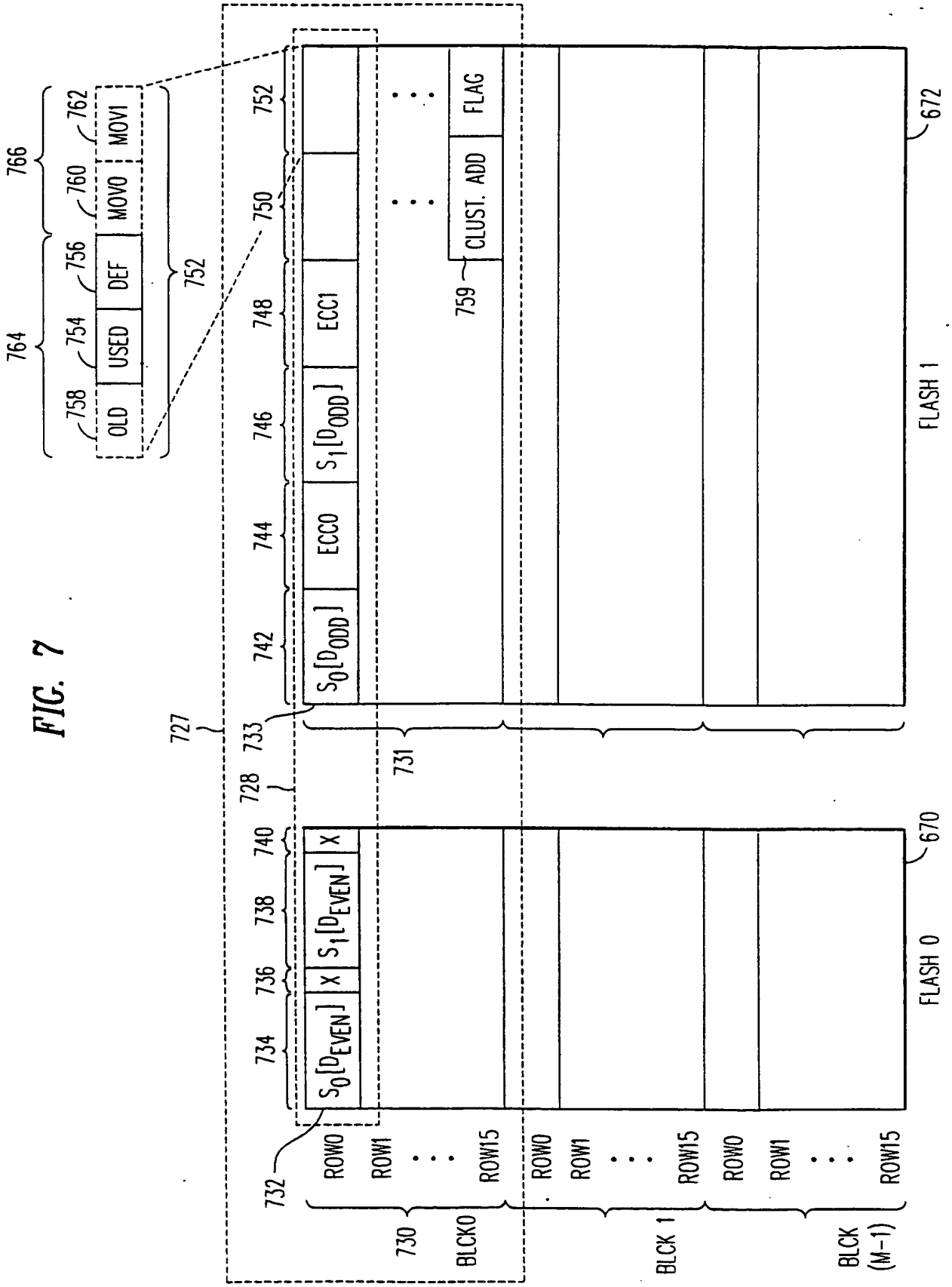


FIG. 7

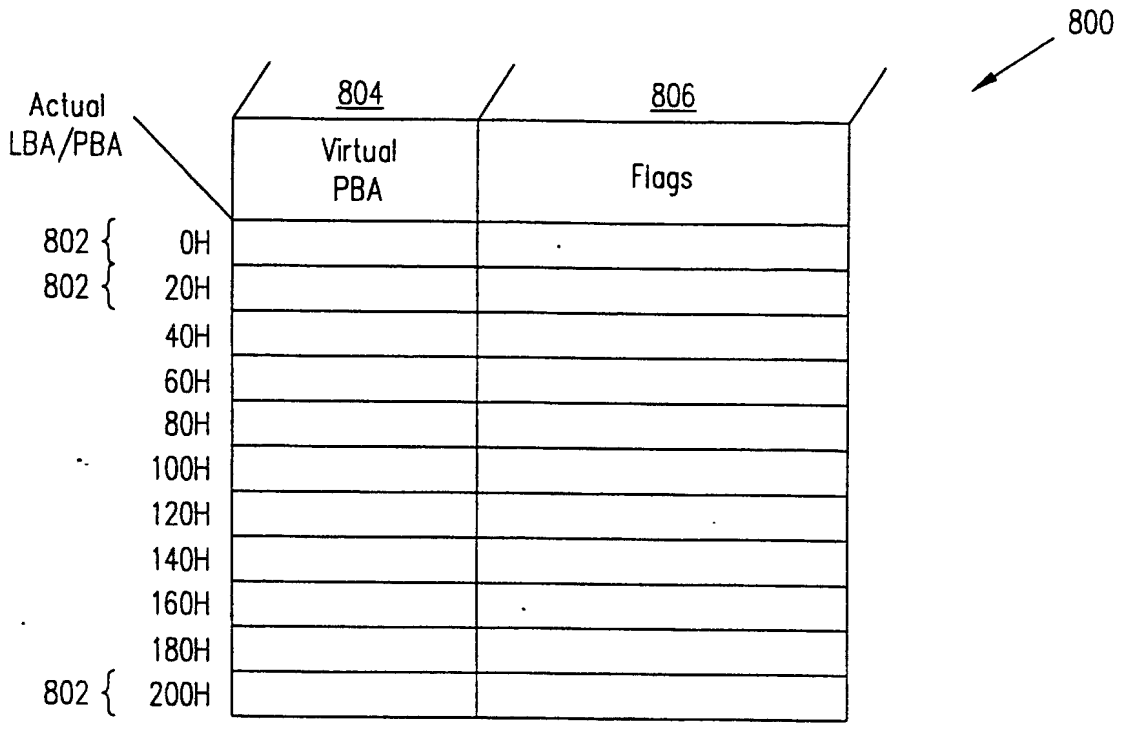


FIG. 8a

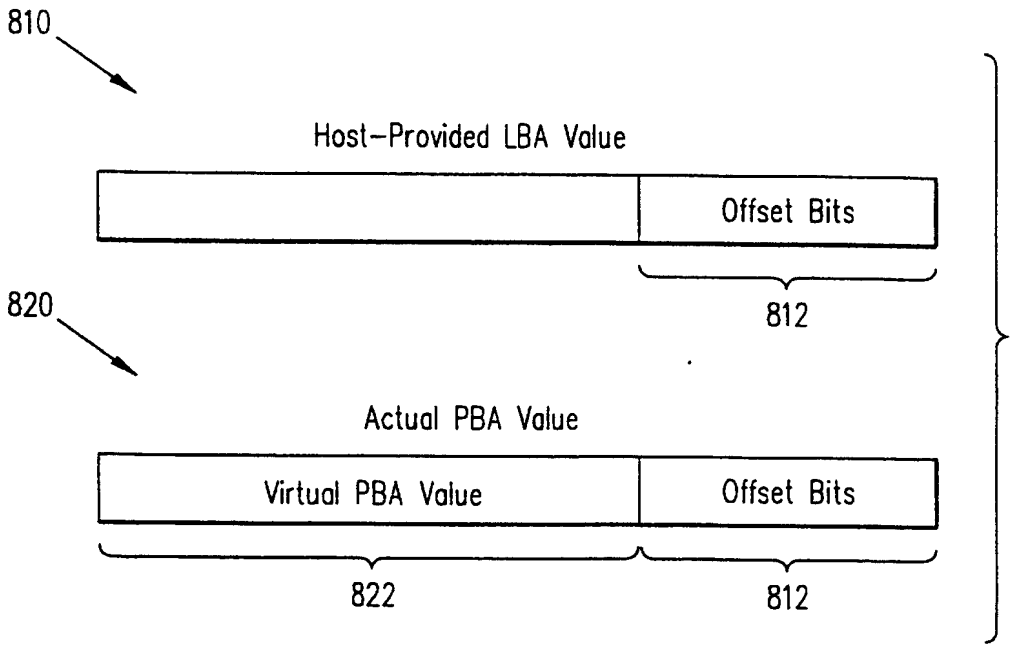


FIG. 8b

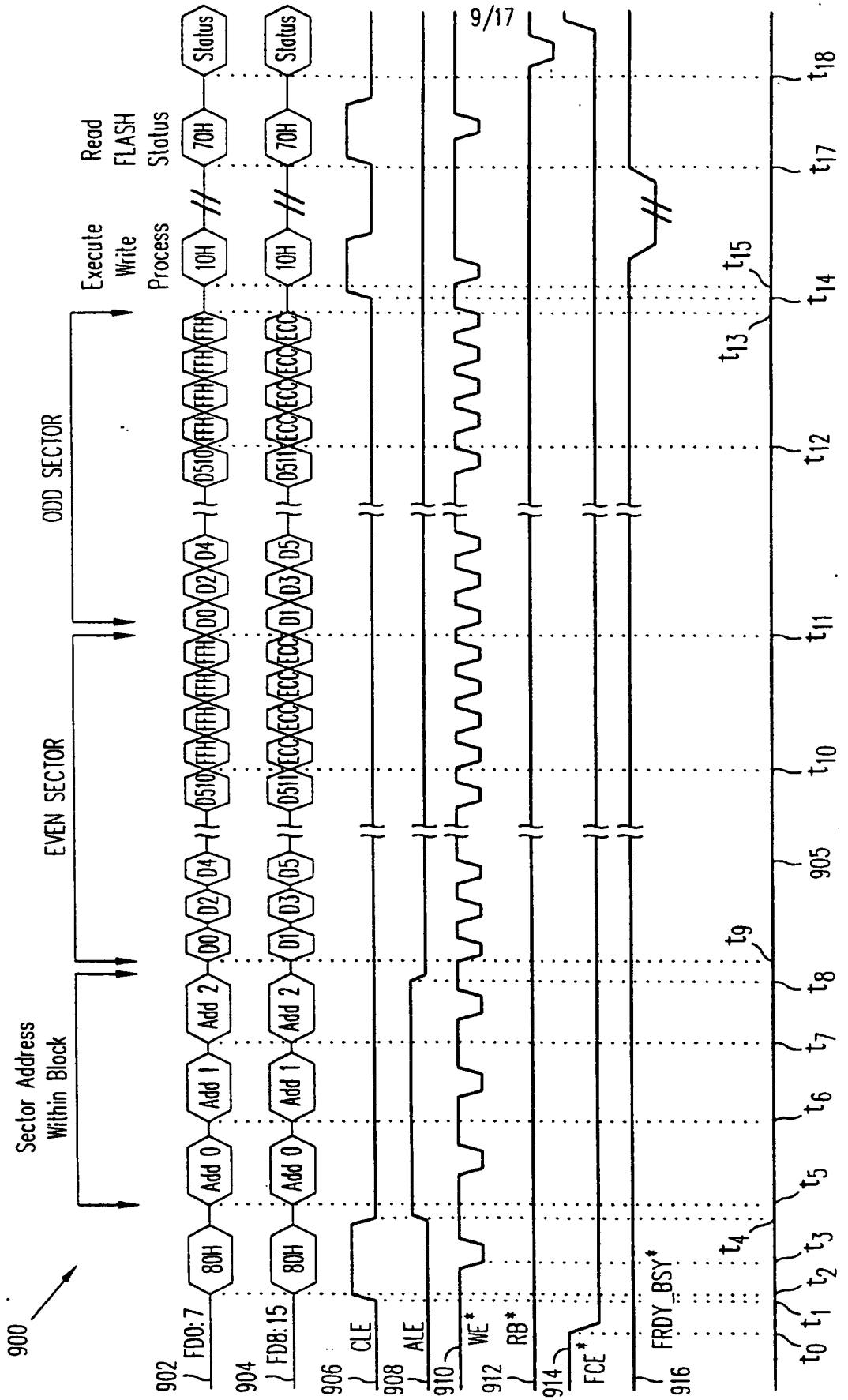


FIG. 9

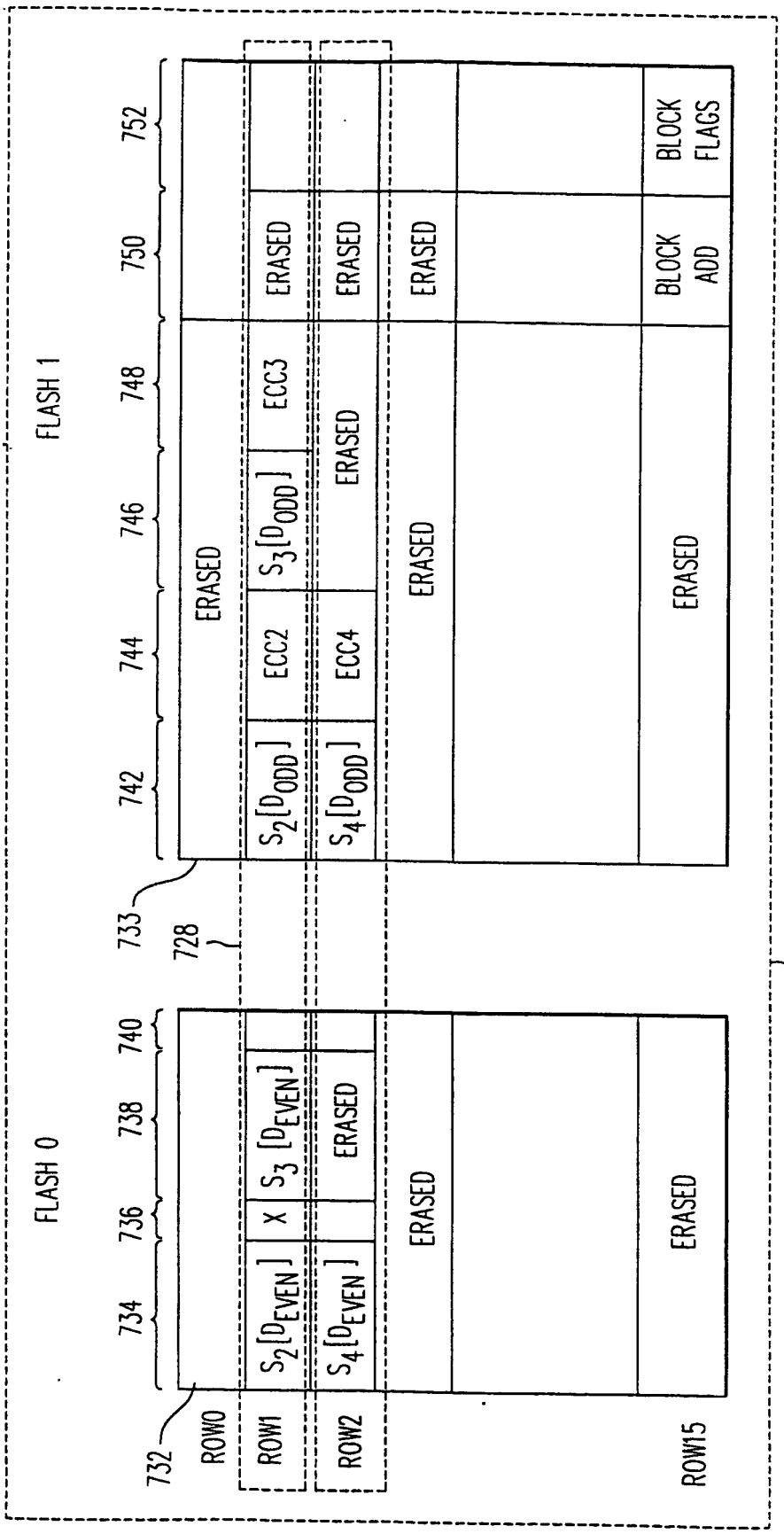


FIG. 10

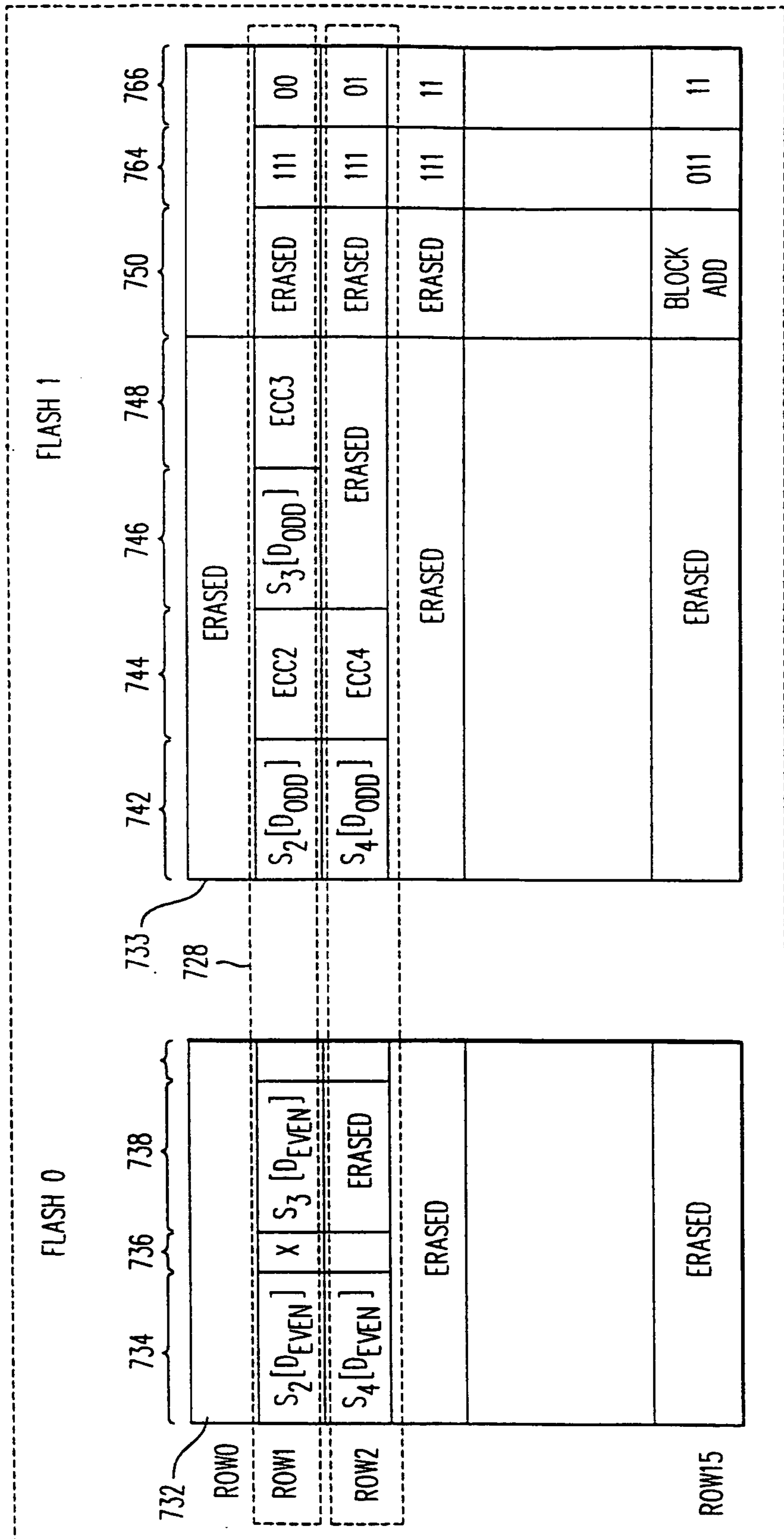

FLASH 0
FLASH 1
732
733
728
727
734
736
738
742
744
746
748
750
764
766
ROW0
ROW1
ROW2
ROW15
S2 [DEVEN]
X
S3 [DEVEN]
S4 [DEVEN]
ERASED
S2 [DODD]
ECC2
S3 [DODD]
ECC3
S4 [DODD]
ECC4
BLOCK ADD
111
011
00
01
11

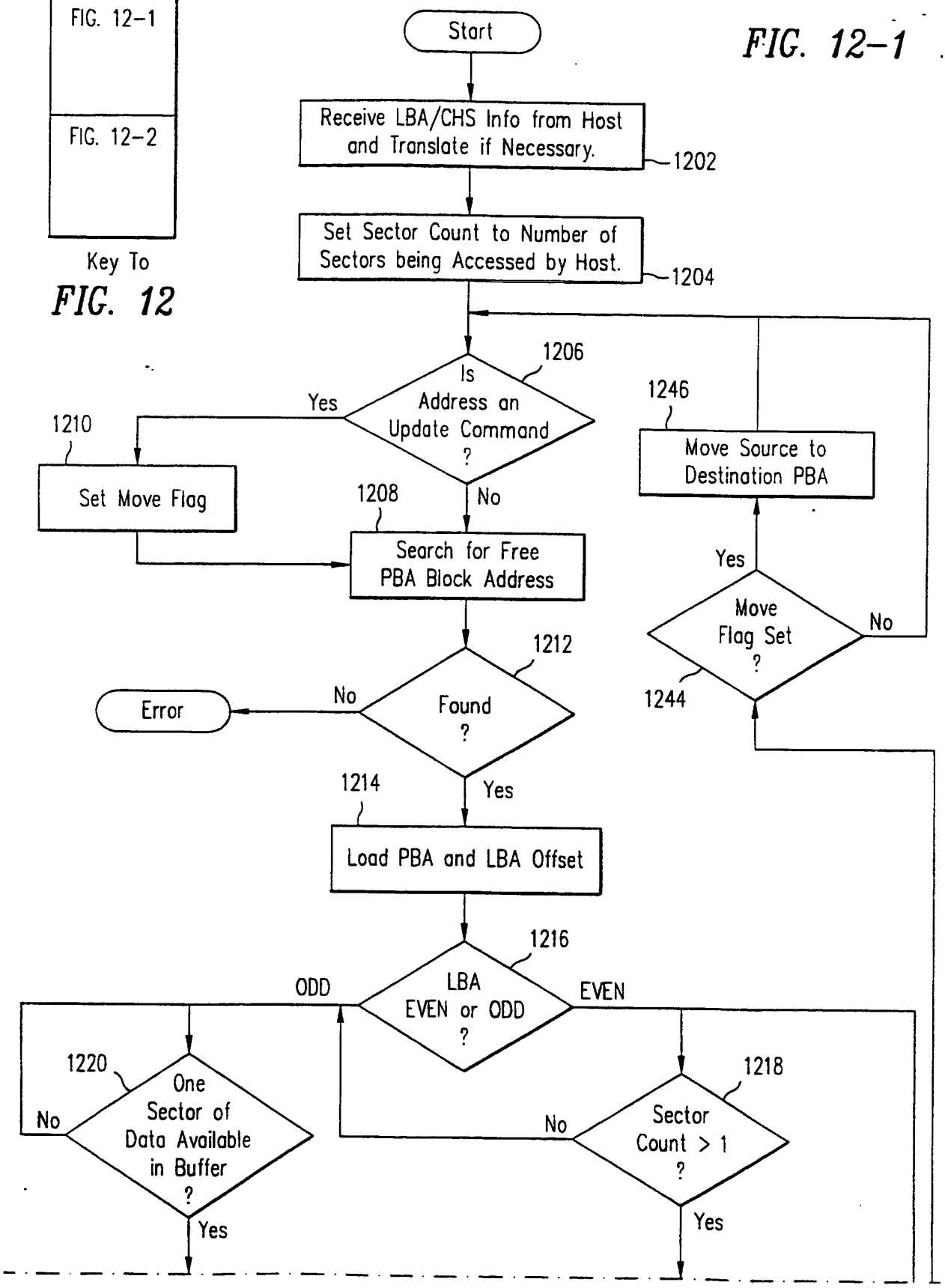
FIG. 11

FIG. 12-1

FIG. 12-2

Key To

FIG. 12



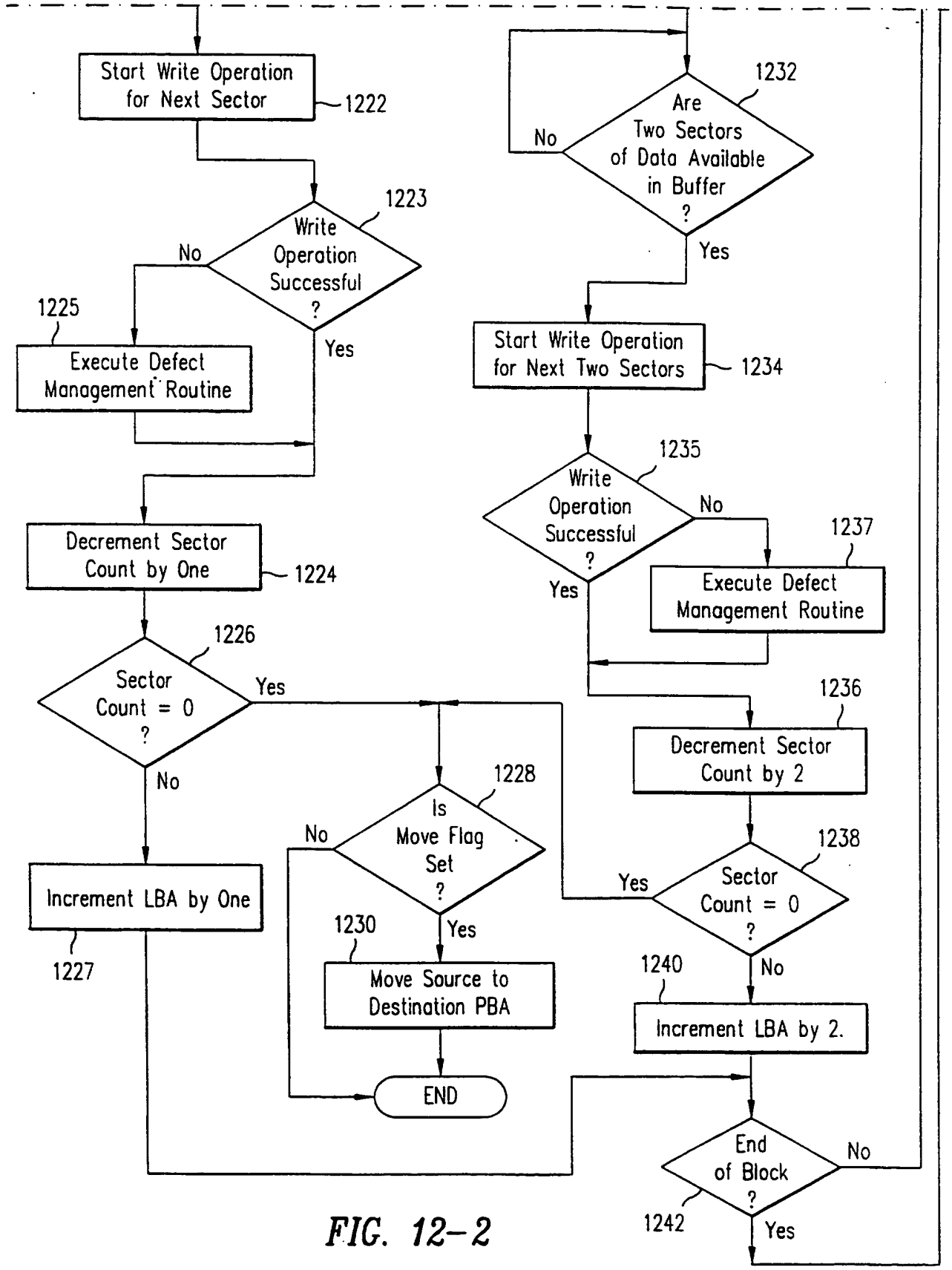


FIG. 12-2

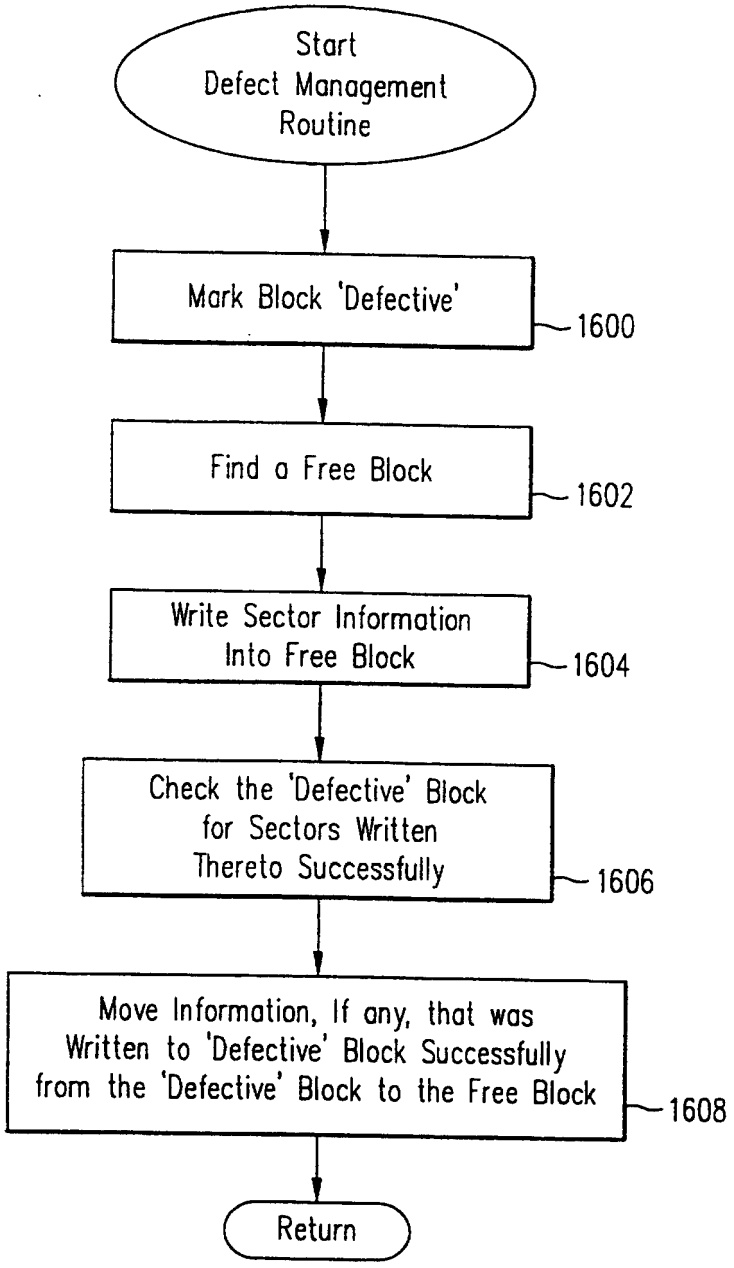


FIG. 12a

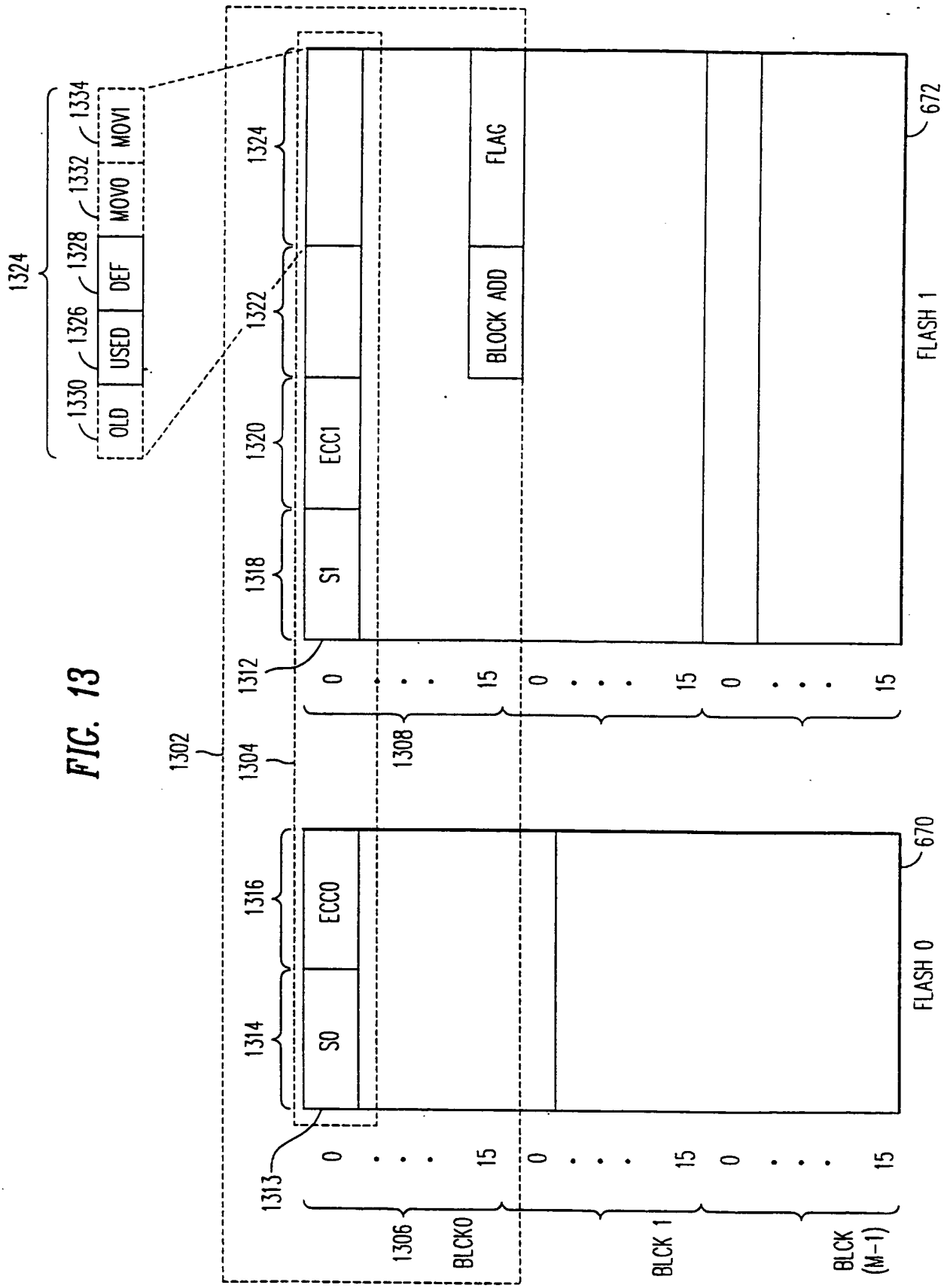


FIG. 13

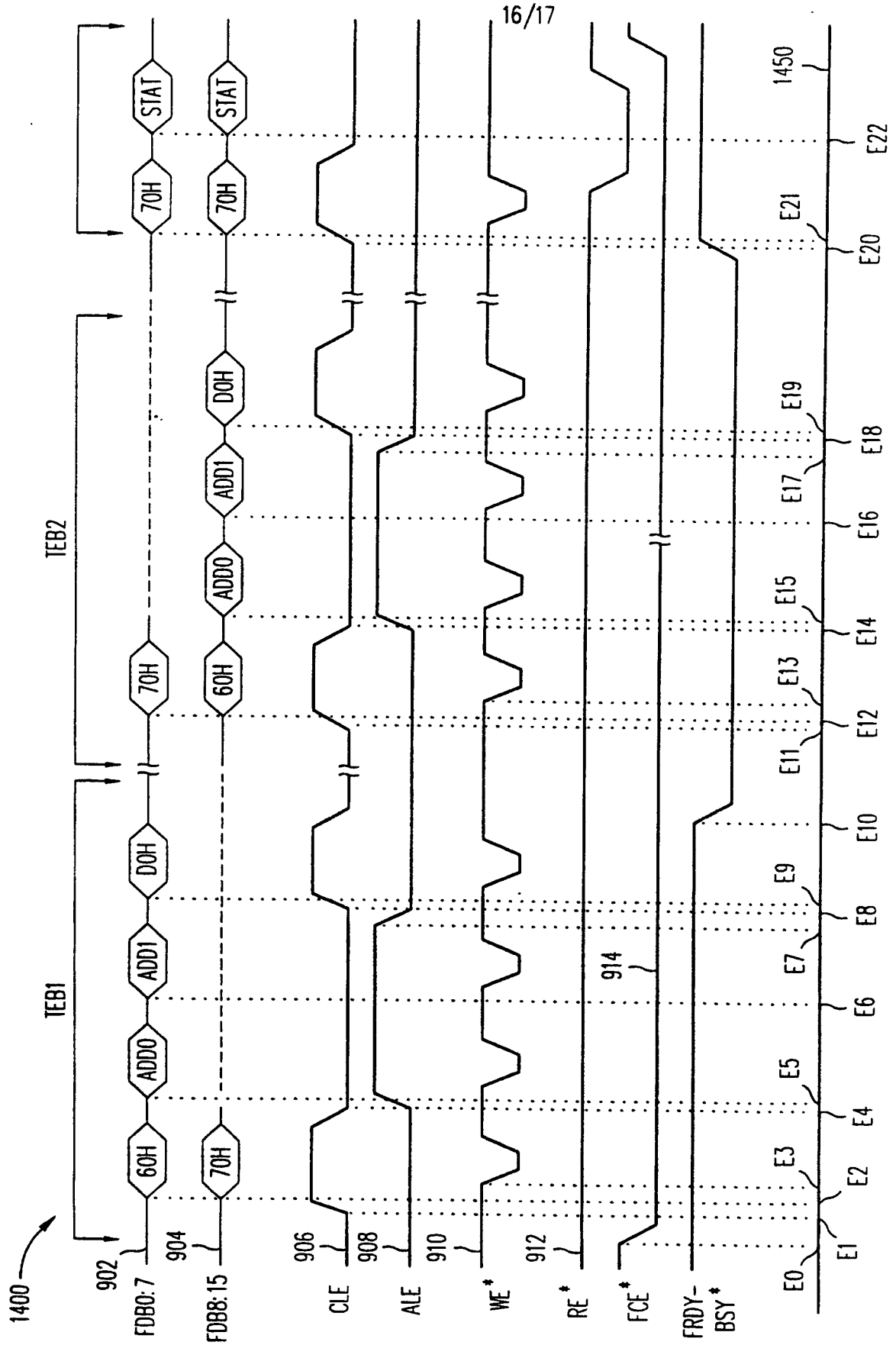


FIG. 14

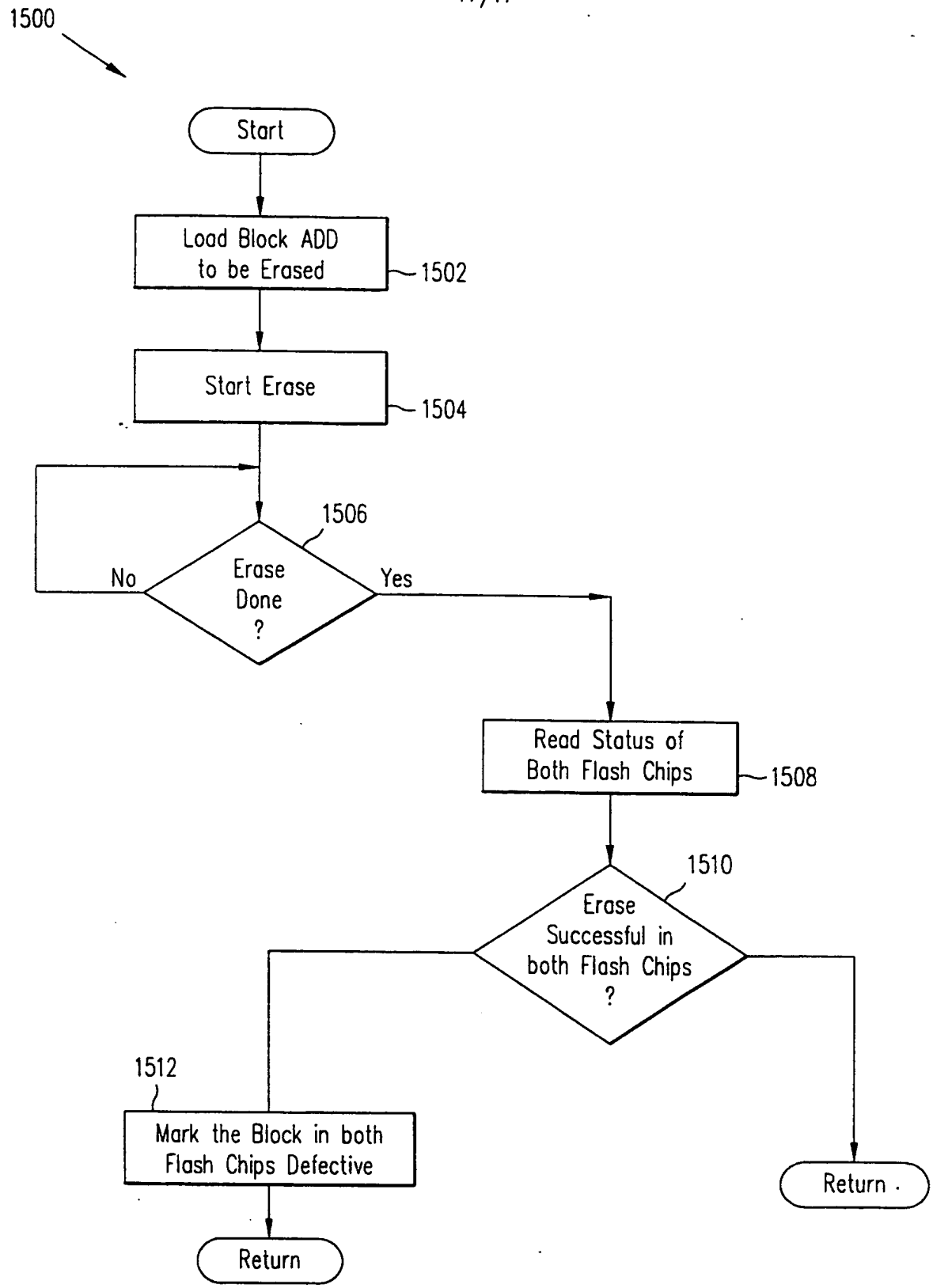


FIG. 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/04247

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) : G06F 12/00, 12/10
US CL. : 711/103, 168, 173, 202, 206
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 711/103, 168, 173, 202, 206

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
APS, DIALOG
search terms: sectors, concurrent or simultaneous, flash memory

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,375,222 A [ROBINSON et al.] 20 December 1994, C 5, L 39-44; C 25, L 1-19; C 24, L 38-68; C 4, L 60-62; C 11, L 24-68; C 12, L 24-68; C 6, L 56-59; C 7, L 7-35	1-7 12-19, 22-25
A,P	US 5,732,208 A [TAMURA et al] 24 March 1998, C 2, L 65-67; C 3, L 1-32;	1, 12-19, 22-25
X,P	US 5,748,528 A [CAMPARDO et al.] 05 May 1998, C 2, L 12-27	12-14, 22-24
A,E	US 5,890,192 A [LEE et al.] 30 March 1999, C 1, L 27-67	1, 12-19, 22-25

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 11 JUNE 1999	Date of mailing of the international search report 17 AUG 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer KIMBERLY James R. Matthews Telephone No. (703) 308-9592

Form PCT/ISA 210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No:
PCT/US99/04247

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,603,001 A [SUKEGAWA et al.] 11 February 1997, C 4, L 13-30	1, 12-19, 22-25
X,P	US 5,745,418 A [MA et al.], 28 April 1998, C 4, L 24-49; C 8, L 50-67; C 9, L 1-35	9-11, 20-21, 26

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*